# Sparse Low Rank Approximation of Multivariate Functions – Applications in Uncertainty Quantification

Prashant Rai

**THÈSE DE DOCTORAT**
**DE L'ÉCOLE CENTRALE DE NANTES**

présentée par

**Prashant RAI**

pour obtenir le grade de

**DOCTEUR DE L'ÉCOLE CENTRALE DE NANTES**

Sujet de la thèse

# Approximation creuses et de faible rang pour les fonctions multivariées: Applications en quantification d'incertitudes

Soutenue à Nantes le 13 Novembre 2014 devant le jury composé de :

| | | |
|---|---|---|
| Olivier LE MAÎTRE | Directeur de Recherche, CNRS, LIMSI, Paris | Rapporteur |
| Bruno SUDRET | Professeur, ETH Zurich, Suisse | Rapporteur |
| Clémentine PRIEUR | Professeur, Université Grenoble Alpes | Examinateur |
| Regis LEBRUN | Ingénieur de Recherche, Airbus Group Innovations | Examinateur |
| Anthony NOUY | Professeur, Ecole Centrale Nantes | Directeur de thèse |
| Mathilde CHEVREUIL | Maître de Conférences, Université de Nantes | Encadrant |

Ecole Centrale de Nantes / CNRS / Université de Nantes
1 Rue de la Noë, 44300 Nantes, France

# Abstract

Uncertainty quantification has been a topic of significant research in computational engineering since early works in stochastic finite element method. Several methods have been proposed based on classical results in analysis and approximation theory. However for problems involving very high stochastic dimension, standard methods are limited by the so called *curse of dimensionality* as the underlying approximation space increases exponentially with dimension. Also, resolution of these high dimensional problems *non intrusively* (when we cannot access or modify model source code), is often difficult with only a partial information in the form of a few model evaluations. Given computation and time resource constraints, methods addressing these issues are needed.

The main idea in this thesis is to perform approximations in low rank tensor formats which incorporates an additional sparsity structure. This methodology exploits both low rank and sparsity structures of high dimensional functions and can thus provide sufficiently accurate approximations with limited information in the form of a few sample evaluations. The number of parameters to estimate in these sparse low rank tensor formats is linear in the stochastic dimension with few non zero parameters that can be estimated efficiently by sparse regularization techniques.

In this thesis, we define sparse low-rank tensor subset in a general setting and propose method based on discrete least-squares for the approximation of a multivariate function from random, noisy-free observations. Sparsity inducing regularization techniques are used within classical algorithms for low-rank approximation. We also detail statistical cross validation techniques for the selection of optimal model parameters (tensor rank, regularization parameters). We then propose algorithms to perform approximation in canonical tensor formats and illustrate them on numerical examples. Due to certain limitations of canonical tensor formats, we propose to extend the method to tree based formats, more specifically Tensor Train (TT) format. Manifolds of low rank TT tensors have favourable properties and dimension of parametrization still scales linearly with stochastic dimension thus making them attractive for high dimensional uncertainty quantification problems. Algorithms based on Density Matrix Renormalization Group combined with sparse regularization for adaptive selection of optimal ranks is proposed. Numerical results indicate considerable advantage of this technique

over other low rank approximation methods.

We finally propose first ideas on combining sparse low rank tensor approximation with clustering and classification techniques to be able to approximate discontinuous or irregular functions. The idea is to cluster the samples in such a way that in each cluster, we can find sufficiently accurate low rank approximation using a few model evaluations. We test this strategy on a toy function and an application example. The results indicate that if the parameters are properly chosen, the method offers considerable advantage than a direct sparse polynomial or low rank approximation.

**Keywords:** Uncertainty quantification, High dimensional approximation, low rank, sparsity, clustering, classification

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Approximation of High Dimensional Stochastic Functions

*In this chapter, we briefly present the basics of probabilistic modelling of uncertainties and their propagation in numerical models in a functional setting. We then present a survey of state of the art methods for high dimensional uncertainty propagation problems. More specifically, a brief description of methods based on structured approximations, i.e. methods that exploit specific structures of high dimensional functions, are presented.*

# Contents

# 1 Uncertainty quantification in scientific computing

Uncertainty quantification has emerged as a crucial field of research over the last decade across various branches of science and engineering. With the exponential increase in available computing resources to model physical phenomena of scientific and engineering interest, there is a growing interest in performing 'virtual' experiments to better understand natural phenomena and predict their outcomes. New findings in research are contributing in the improvement of available numerical tools. These tools are increasingly taking into account the uncertainties in representation of numerical models. Consequently, uncertainty quantification and propagation in physical systems appear as a critical path for the improvement of the prediction of their response.

Uncertainty in prediction is usually classified under two categories: intrinsic uncertainties, associated with the natural variability of the considered physical phenomenon, and epistemic uncertainties which result from a lack of knowledge and hence, are reducible. However, due to the complexity of physical phenomena or lack of observations, addressing epistemic uncertainty also appears essential in order to improve predictability of the model.

## 1.1 General framework of uncertainty propagation

The probabilistic framework is the most well established way to model uncertainty. The general methodology of an uncertainty study in this framework involves 3 steps as described below:

1. **Specification of the case study**. It involves identification of the model or series of *model*, associated *input variables*, *quantities of interest* and criteria that should be used to assess the physical system under consideration. A quantity of interest denotes the output variable on which uncertainty is to be quantified. A model denotes a mathematical function that enables the computation of the quantity of interest when giving several input variables. The associated criteria could be probability of exceeding a threshold, measure of central dispersion, quantiles, sensitivity indices (e.g. Sobol indices for global sensitivity analysis).

2. **Quantifying the sources of uncertainty**. It involves parametrization of input variables as random variables. In some cases, possible dependencies between the input variables also needs to be investigated thus requiring a multidimensional analysis. The description of the time/space variability of parameters requires the introduction of random processes.

3. **Propagating the uncertainty**. This step consists of propagating the uncertainty in the input through the model, i.e. characterizing the random re-

sponse appropriately with respect to the assessment criteria defined in the first step. Methods to carry out this task are usually classified as *intrusive* and *non-intrusive*. Intrusive methods require the modification of the model to incorporate probabilistic framework whereas in non-intrusive methods classical deterministic numerical codes can be used.

## 1.2 Probabilistic modelling of uncertainties

In the probabilistic framework, modelling the uncertainties consists of defining a probability space $(\Theta, \mathcal{B}, P)$, where $\Theta$ denotes the space of elementary events, $\mathcal{B}$ denotes a sigma algebra defined on $\Theta$ and $P$ is a probability measure. The model response is then a random variable, with value in a certain function space, that verifies almost surely a set of equations formally denoted

$$\mathcal{F}(u(\theta), \theta) = 0. \tag{1.1}$$

The choice of the probability space is crucial in the modelling step. The mathematical study of problem (1.1) depends on the nature of uncertainty which determines the representation of the solution and solution methods.

## 1.3 Discretization of uncertainties

We consider that the uncertainty in the input can be correctly modelled with a finite set of random variables $\xi : \theta \in \Theta \mapsto \xi(\theta) \in \Xi$, defining a new finite dimensional probability space $(\Xi, \mathcal{B}_\Xi, P_\xi)$, where $\Xi = \xi(\Theta)$, $\mathcal{B}_\Xi$ is a $\sigma$-algebra on $\Xi$ and $P_\xi$ is a probability measure associated with $\xi$. This case is encountered when the uncertainties in the model are represented as real valued random variables or stochastic fields. In case of stochastic fields, spectral decomposition techniques (such as Karhunen-Loeve decomposition) can be used for discretization. The model output is then searched as a function of $\xi$ satisfying almost surely the set of equations

$$\mathcal{F}(u(\xi), \xi) = 0. \tag{1.2}$$

# 2 Functional representation and choice of approximation bases

Several methods are available in the literature for solving (1.2). The selection of a particular method depends on the problem under consideration, the quantity of interest (e.g. statistical moments, probability of an event etc.), availability of computation resources etc. In this section, we introduce methods based on functional representation of random variables. A function of random variables $u(\xi)$ can be written under the form

$$u(\xi) = \sum_{\alpha \in \Lambda} u_\alpha \phi_\alpha(\xi), \tag{1.3}$$

where $u_\alpha$ are the coefficients on a certain basis $\{\phi_\alpha\}_{\alpha\in\Lambda}$ and $\Lambda$ is an index set. A sufficiently accurate representation of $u(\xi)$ under the form (1.3) enables rapid post-processing studies by performing fast evaluations of $u(\xi)$. Since the work of Ghanem and Spanos [35], there was an increase in interest in this type of approaches often referred to as *"spectral stochastic approaches"*.

For a given physical model, when uncertainties are characterized by a finite set of random variables $\xi = (\xi_1, \ldots, \xi_d)$, we work in associated finite dimensional probability space $(\Xi, \mathcal{B}_\Xi, P_\xi)$, where $\Xi \subset \mathbb{R}^d$. The quantity of interest is then interpreted as a random variable defined on $(\Xi, \mathcal{B}_\Xi, P_\xi)$. A large class of problems often encountered in physical applications can be characterised by finite variance of the uncertain inputs and model outputs. This leads to introduce the space of square integrable functions $L^2$.

$$L^2(\Xi, dP_\xi) = \left\{ v : \xi \in \Xi \mapsto v(\xi) \in \mathbb{R};\ E(v^2) := \int_\Xi v(\mathbf{y})^2 dP_\xi(\mathbf{y}) < \infty \right\}, \tag{1.4}$$

which is a Hilbert space endowed with the following inner product

$$\langle v, w \rangle_{L^2(\Xi, dP_\xi)} = E(vw) = \int_\xi v(\mathbf{y}) w(\mathbf{y}) dP_\xi(\mathbf{y}), \tag{1.5}$$

where $E$ is the mathematical expectation. A Hilbertian basis $\{\phi_\alpha\}_{\alpha\in\Lambda}$ of $L(\Xi, dP_\xi)$ is a complete set of orthonormal functions such that:

$$\langle \phi_\alpha, \phi_\beta \rangle_{L^2(\Xi, dP_\xi)} = \delta_{\alpha\beta} \tag{1.6}$$

Each function $u(\xi) \in L^2(\Xi, dP_\xi)$ admits a unique decomposition on such a basis:

$$v(\xi) = \sum_{\alpha\in\Lambda} v_\alpha \phi_\alpha(\xi), \tag{1.7}$$

$$v_\alpha = \langle v, \phi_\alpha \rangle_{L^2(\Xi, dP_\xi)} = E(v\phi_\alpha). \tag{1.8}$$

In cases where random variables $\xi$ are mutually independent, approximation basis can be obtained from tensorization of univariate bases. Let us denote $(\Xi_k, \mathcal{B}_{\xi_k}, P_{\xi_k})$ as the probability space associated with random variable $\xi_k$. We have

$$\Xi = \Xi_1 \times \ldots \times \Xi_d, \tag{1.9}$$

$$P_\xi = \bigotimes_{k=1}^{d} P_{\xi_k}, \tag{1.10}$$

$$L^2(\Xi, dP_\xi) = \bigotimes_{k=1}^{d} L^2(\Xi_k, dP_{\xi_k}). \tag{1.11}$$

A basis of $L^2(\Xi, dP_\xi)$ can then be obtained by tensorization of basis of $L^2(\Xi_k, dP_{\xi_k})$. Denoting $\{\phi^k_{\alpha_k}\}_{\alpha_k \in \Lambda^k}$ a basis of $L^2(\Xi_k, dP_{\xi_k})$, we let

$$\phi_\alpha(y) = \phi^1_{\alpha_1}(y_1) \cdots \phi^d_{\alpha_d}(y_d), \tag{1.12}$$

with $\alpha = (\alpha_1, \ldots, \alpha_d) \in \Lambda = \Lambda^1 \times \cdots \times \Lambda^d$. If the basis functions $\{\phi^k_{\alpha_k}\}_{\alpha_k \in \Lambda^k}$ are orthonormal with respect to the natural inner product in $L^2(\Xi_k, dP_{\xi_k})$, basis functions $\{\phi_\alpha\}_{\alpha \in \Lambda}$ are also orthonormal. A function of $\xi = (\xi_1, \ldots, \xi_d)$ random variables can then be represented as

$$u(\xi_1, \ldots, \xi_d) = \sum_{\alpha_1 \in \Lambda^1}^{n} \cdots \sum_{\alpha_d \in \Lambda^d}^{n} u_{\alpha_1 \ldots \alpha_d} \phi^1_{\alpha_1}(\xi_1) \cdots \phi^d_{\alpha_d}(\xi_d), \; u_{\alpha_1 \ldots \alpha_d} \in \mathbb{R}. \tag{1.13}$$

Thus, the total number of parameters $N = \#\Lambda$ (i.e. the number of coefficients of the multidimensional basis) needed to characterize the approximation of $u$ is given by $N = \prod_{k=1}^d \#\Lambda^k$. Suppose $\#\Lambda^k = n$ for all $k$. For high dimensional functions, $d$ is large and hence $N = n^d$ increases exponentially with $d$. This is referred to as the *curse of dimensionality*. Under the assumption $u \in \mathcal{C}^s$, the accuracy $\epsilon$ of approximation (1.13) is expected to be such that $\epsilon = O(n^{-s}) = O(N^{-s/d})$ i.e. to achieve approximation accuracy $\epsilon$, the number of parameters needed is $N = O(\epsilon^{-d/s})$ (exponential increase with $d$). Thus global smoothing is not enough to beat the curse of dimensionality. Note that there is also a notion of curse of dimensionality for high dimensional integration [42]. However, in this thesis, we are concerned with the curse of dimensionality associated to approximation.

A natural choice of basis functions are polynomial basis. The space of multidimensional polynomials with partial degree $p$ defined on $\Xi \subset \mathbb{R}^d$ is denoted

$$\mathbb{Q}_p(\Xi) = \text{span}\left\{ \prod_{k=1}^d y_k^{\alpha_k} : \alpha \in \mathbb{N}^d, |\alpha|_\infty := \max_{k \in \{1 \ldots d\}} \alpha_k \leq p \right\}, \tag{1.14}$$

with $\dim(\mathbb{Q}(\Xi))=(p+1)^d$. The space of multidimensional polynomials of total degree $p$ defined on $\Xi \subset \mathbb{R}^d$ is defined by:

$$\mathbb{P}_p(\Xi) = \text{span}\left\{ \prod_{k=1}^d y_k^{\alpha_k} : \alpha \in \mathbb{N}^d, |\alpha|_1 := \sum_{k=1}^d \alpha_k \leq p \right\} \tag{1.15}$$

with $\dim(\mathbb{P}_p(\Xi)) = \frac{(d+p)!}{d!p!}$. If $\Xi = \Xi_1 \times \cdots \times \Xi_d$, $\mathbb{Q}_p(\Xi)$ is a full tensorization of unidimensional polynomial spaces of degree $p$:

$$\mathbb{Q}_p(\Xi) = \mathbb{Q}_p(\Xi_1) \otimes \cdots \otimes \mathbb{Q}_p(\Xi_d). \tag{1.16}$$

The space $\mathbb{P}_p(\Xi)$ can be considered as a sparse tensorization of polynomial spaces $\mathbb{Q}_p(\Xi_k)$:

$$\mathbb{P}_p(\Xi) = \sum_{\alpha \in \mathbb{N}^d, |\alpha|_1 \leq p} \mathbb{Q}_{\alpha_1}(\Xi_1) \otimes \cdots \otimes \mathbb{Q}_{\alpha_d}(\Xi_d). \tag{1.17}$$

In polynomial chaos representation, we use classical orthogonal polynomial basis of $L^2(\Xi, dP_\xi)$. For independent random variables, basis are obtained by a sparse tensorization of polynomial bases of $L^2(\Xi, dP_{\xi_k})$. The orthogonal family of polynomials in $L^2(\Xi_k, dP_{\xi_k})$ for a given probability measure of $P_{\xi_k}$ are uniquely defined. For a general introduction to orthogonal polynomials for different probability measures, see [82].

Another approach is to introduce a basis of interpolating polynomials [1, 33]. Considering independent random variables, on each stochastic dimension, we introduce a set of points $\mathcal{L}_k = \{y_{k,n}\}_{n=0}^p \subset \Xi_k$ and define the associated interpolation basis $\{\phi_n^k\}_{n=0}^p$;

$$\phi_n^k \in \mathbb{Q}_p(\Xi_k),\ \phi_n^k(y_k, l) = \delta_{nl}. \tag{1.18}$$

Particular interpolation points can be given by the roots of the classical orthogonal polynomial of degree $(p+1)$, i.e. $(p+1)$ points associated with measure $dP_{\xi_k}$. This choice leads to orthogonal interpolation functions:

$$\langle \phi_n^k, \phi_l^k \rangle_{L^2(\Xi_k, dP_{\xi_k})} = E(\phi_n^k \phi_l^k) = \int_{\Xi_k} \phi_n^k(y)\phi_l^k(y) dP_{\xi_k}(y) = \delta_{nl}\omega_n, \tag{1.19}$$

where $\omega_n$ denote Gauss quadrature weights. Basis of $\mathbb{Q}_p(\Xi)$ can be obtained by a full tensorization of interpolation bases of $\mathbb{Q}_p(\Xi_k)$ which are interpolation basis on a multidimensional grid obtained by full tensorization of unidimensional grids $\mathcal{L}_k$. Basis of $\mathbb{P}_p(\Xi)$ can be obtained by a sparse tensorization of unidimensional interpolation basis using Smolyak construction [33].

For representing non-smooth functions, piecewise polynomial functions [26, 78] defined on a partition of $\Xi$ can be used. Adaptivity on this basis can be obtained by refining partition or increasing approximation degree [51, 79]. Another common way consists in using polynomial multi-wavelets basis [49, 48] that allows for a multi-scale representation of functions.

# 3 Methods for Computing Approximations

Several methods exist in the literature for the determination of expansion coefficients in (1.7). In the following, we briefly introduce two specific approaches that only use sample evaluations of functions and therefore do not require specific implementation and the Galerkin projection method.

## 3.1 $L^2$ Projection

The $L^2$ projection method consists of defining the approximation (1.7) as the projection of $u$ on the sub-space of $L^2(\Xi, dP_\xi)$ spanned by functions $\{\phi_\alpha\}_{\alpha\in\Lambda_P}$. Thus

the coefficients $u_\alpha$ are given by

$$u_\alpha = \langle u, \phi_\alpha \rangle_{L^2(\Xi, dP_\xi)} = \int_\Xi u(y)\phi_\alpha(y) dP_\xi(y). \tag{1.20}$$

The computation of coefficients then requires the evaluation of an integral on $\Xi$ with respect to measure $dP_\xi$. One of the widely used methods for evaluating these integrals is based on tensorization of classical quadratures [64]. Suppose that a quadrature rule is defined on each stochastic dimension

$$Q^k(u) = \sum_{j=1}^{J_k} u(y_{k,j})\omega_{k,j} \approx \int_{\Xi_k} u(y) dP_{\xi_k}(y). \tag{1.21}$$

A quadrature in dimension $d$ can be obtained by full tensorization of unidimensional quadratures:

$$Q_J = Q^1 \otimes \ldots \otimes Q^d,$$

with

$$Q_J(u) = \sum_{j_1=1}^{J_1} \ldots \sum_{j_d=1}^{J_d} u(y_{1,j_1}, \ldots, y_{d,j_d})\omega_{1,j_1}, \ldots, \omega_{d,j_d}.$$

For quadratures with $J_k = n$ points on each dimension, one obtains a total number of points $J = n^d$, which increases exponentially with the stochastic dimension thus manifesting the curse of dimensionality. For a function of class $\mathcal{C}^r$, the integration error verifies

$$|E(u) - Q_J(u)| = O(J^{-(2r-1)/d}). \tag{1.22}$$

An alternative approach is Smolyak sparse grids [34, 55, 63, 67] that drastically reduces the number of integration points when dealing with high stochastic dimension $d$. It requires the definition of a sequence of quadratures $\{Q_j^k\}_{j=1}^l$ on each dimension, where $j$ denotes the level of quadrature. Thus Smolyak sparse grid is still based on tensor product construction but it avoids the use of high-level quadratures on several dimensions simultaneously. A level $l$ quadrature in dimension $d$ is obtained by the following tensorization formula:

$$Q_J^l = \sum_{j \in \mathbb{N}^d, l \leq |j| \leq l+d-1} (-1)^{l+d-1-|j|} \begin{pmatrix} l-1 \\ |j|-l \end{pmatrix} Q_{j_1}^1 \otimes \cdots \otimes Q_{j_d}^d. \tag{1.23}$$

Here, if $Q_j^k$ denote $j$-points quadrature, one obtains a total number of integration points $J = O(\frac{2^l}{l} d^l)$. The integration error depends on the smoothness of function $u$. For a $r$-times differential function $u$, the integration error is given by:

$$|E(u) - Q_J^l(u)| \sim O(J^{-r} log(J)^{(m-1)(r+1)}), \tag{1.24}$$

that gives a better convergence rate compared to full tensorization. Tensorization rules are more advantageous when nested quadrature points are used in each dimension i.e. the set of integration points of a quadrature $Q_j^k$ is included in that of $Q_{j+1}^k$. This leads to significant cost reduction. Such tensor grids are called as *sparse grids* [11]. One can also modify the tensorization to weight various dimensions in accordance with their influence. Such grids are called *adaptive sparse grids* [10].

The Monte Carlo technique [12, 66] can be used for quadrature with $Q$ quadrature points sampled randomly and weights taken equal to $\omega = \frac{1}{Q}$. However, convergence rate is very slow. Improved convergence rates can be achieved by using more efficient stochastic sampling techniques such as Latin hypercube sampling or quasi-Monte Carlo method [12, 53, 71].

## 3.2 Interpolation

Methods based on interpolation [1] consists in choosing an interpolation basis $\{L_\alpha\}_{\alpha\in\Lambda}$ on a set of points $\{y_\alpha\}_{\alpha\in\Lambda}$. Coefficients of the decomposition of $u(\xi)$ is obtained by solving a deterministic problem associated with $y_\alpha$. This method is often called stochastic collocation. For high dimensional functions, Smolyak tensorization of unidimensional interpolation can be used [33]. The interpolation properties of multidimensional polynomials is preserved if nested interpolation grids are used in the Smolyak tensorization. Recent work on interpolation techniques can be found in [20].

## 3.3 Galerkin Projection

Let us consider the stochastic partial differential equation with random input parameters given by:

$$\mathcal{J}(x,t,\xi;u) = f(x,t,\xi),$$

where $u = u(x,t,\xi)$ is the solution and $f(x,t,\xi)$ is the source term. $\mathcal{J}$ is a general differential operator that may contain spatial derivatives, time derivatives, and linear or non-linear terms. The $\xi$ denotes the dependence on some random input parameter, which may be equation parameters, boundary conditions, input conditions, or even the domain. We represent the solution as functional representation,

$$u = u(x,t,\xi) = \sum_{\alpha\in\Lambda_P} u_\alpha(x,t)\phi_\alpha(\xi), \tag{1.25}$$

where $\Lambda_p \subset \Lambda$. Substituting this expansion into the differential equation, we define the approximation by cancelling the projection of the residual onto the approximation space

$$\langle \mathcal{J}(x,t,\xi;\sum_{\alpha\in\Lambda_P} u_\alpha\phi_\alpha), \phi_\beta\rangle = \langle f(x,t,\xi),\phi_\beta\rangle, \forall\alpha\in\Lambda_P. \tag{1.26}$$

Since we take the projection, the error in the approximate solution is orthogonal to the space spanned by $\phi_\alpha$. The stochastic differential equation reduces to system of $\#\Lambda_P$ coupled deterministic differential equations for the coefficients of polynomial chaos expansion.

Curse of dimensionality manifests in all the methods presented above (when standard approximation spaces are used) and hence we look for methods that exploit specific structures of high dimensional functions. These are discussed in the next section.

# 4 Methods using structured approximations

Exploiting smoothness property is not sufficient to approximate high dimensional functions (see section above). Novak [57] showed that approximation problem remains intractable even for $u \in \mathcal{C}^\infty$ with uniformly bounded derivatives i.e. $\sup_\alpha \|D^\alpha u\|_\infty < \infty$. In the following section, we briefly describe different class of methods that exploit certain structures of high dimensional functions thus enabling to circumvent the curse of dimensionality.

## 4.1 Low order interactions

These methods approximate a high dimensional function based on the assumption that higher order interaction terms have negligible effect on the model response. In this approach, one employs the hierarchical expansion of the model response $u(\xi)$ under the form:

$$u(\xi) = u_0 + \sum_{1\le i\le d}^{d} u_i(\xi_i) + \sum_{1\le i<j\le d} u_{i,j}(\xi_i,\xi_j) + \ldots \\ \sum_{1\le i_1<\ldots<i_s\le d} u_{(i_1,\ldots,i_s)}(\xi_{i_1},\ldots,\xi_{i_s}) + \ldots + u_{1,2,\ldots,d}(\xi_1,\ldots,\xi_d), \quad (1.27)$$

which is unique upto orthogonality constraints on the functions in the expansion. The constant function $u_0$ is the mean response of the model, first-order univariate functions $u_i(\xi_i)$ represent independent contributions of the individual parameters, also called main effects. Similarly, the bivariate functions $u_{i,j}(\xi_i,\xi_j)$ quantify the interactions of $\xi_i$ and $\xi_j$ on the response $u$ and so on. Note that (1.27) can be simply obtained from (1.7) such that

$$u(\xi) = \sum_{n\ge 0} \sum_{|\alpha|_0=n} u_\alpha \phi_\alpha(\xi), \quad (1.28)$$

where $\sum_{|\alpha|_0=n} u_\alpha \phi_\alpha$ coincides with the $n$-th term of expansion (1.27). Truncating the series at a certain level $n$ yields an approximation $\sum_{\alpha\in\Lambda_n} u_\alpha\phi_\alpha$ with

$$\Lambda_n = \{\alpha \in \Lambda : |\alpha|_0 \leq n\}, \tag{1.29}$$

the index set with interaction order at most $n$. The cardinality of $\Lambda_n$ increases slowly with $d$ for small values of $n$. This a priori choice of indices delays the introduction of high-order interactions. Approximation of $u(\xi)$ in $\mathcal{M}_n = \left\{u(\xi) = \sum_{\alpha\in\Lambda_n} u_\alpha\phi_\alpha(\xi); u_\alpha \in \mathbb{R}\right\}$ leads to favour the main effects and low order interaction, which are more likely to be significant than the high order interactions. In practice, we use $\Lambda_{p,n} = \{\alpha : |\alpha|_\infty \leq p, |\alpha|_0 \leq n\}$ (bounded degree polynomial expansions). An application of this technique using the polynomial chaos basis is shown in [9].

## 4.2 Low effective dimensionality

In this section, methods that exploit low effective dimensionality of the function are described. The fundamental idea in this approach is that although the function may have a large number of random inputs, it is mainly dependent on only few of these variables i.e.

$$u(\xi) \approx u(\xi_K) \tag{1.30}$$

where $K \subset \{1, \ldots, d\}$ and $\#K$ is small. In the following, we outline three variants of this approach. First method is based on screening the important inputs using Global sensitivity analysis. Second method is based on basis adaptation for characterization of subspaces associated with low dimensional quantities of interest [75]. The third method determines the direction of strongest variability using gradients of $u(\xi)$ and constructs an approximation on a low dimensional subspace of the input parameter space [22].

**Variance based method for dimension reduction [69].** In this method, sensitivity analysis is performed to determine which variables have the most influence on the model response. With a ranking of inputs, response surface are then constructed that "concentrates" the approximation on the most influential variables. A standard approach is to use variance based global sensitivity analysis to select the most influential inputs. Let $D = \{1, \ldots, d\}$. For a non empty subset $K \subset D$, we denote $\xi_K = \{\xi_k\}_{k\in K}$. Suppose that $u$ admits the following decomposition

$$u(\xi) = u_\emptyset + \sum_{K\subset D} u_K(\xi_K), \tag{1.31}$$

with $E(u_K) = 0$ for all non empty $K \subset D$ and $E(u_K u_{K'}) = 0$ for $K \neq K'$. Note that this decomposition coincides with (1.28) when writing $\sum_{k\subset D} = \sum_{n\geq 1} \sum_{\substack{K\subset D \\ \#K=n}}$

so that $\sum_{\substack{K\subset D\\ \#K=n}} u_K = \sum_{|\alpha|_0=n} u_\alpha \phi_\alpha$. With this representation, the variance of $u$ can be decomposed as

$$V(u) = \sum_{K\subset D} V(u_K(\xi_K)). \tag{1.32}$$

We define Sobol index $S_K$ as the contribution of $u_K(\xi_K)$ to the total variance

$$S_K = \frac{V(u_K)}{V(u)}, \ \sum_{K\subset D} S_K = 1. \tag{1.33}$$

Thus $S_K$ can be seen as a measure of variability in the output contributed by variables $K \subset D$ . For a single random variable $\xi_k$, the first order Sobol index [70, 43] $S_k$ and the total sensitivity index $S_k^T$ are given by

$$S_k = \frac{V(u_k)}{V(u)}, \ S_k^T = \sum_{K\supset k} S_K. \tag{1.34}$$

We can then keep only those parametric inputs for uncertainty quantification studies for which the Sobol indices are significant. Note that if one has representation of $u(\xi)$ on polynomial chaos basis, Sobol indices can be directly estimated as shown in [73].

**Basis adaptation [75]:** In this method, the function is searched under the form

$$u(\xi) \approx v(g(\xi)) \tag{1.35}$$

where $g(\xi) : \mathbb{R}^d \to \mathbb{R}$ is a linear function of variables $\xi = (\xi_1, \ldots, \xi_d)$. Suppose $\xi$ are independent gaussian random variables. Here, we exploit the mathematical structure of Gaussian Hilbert spaces. The method consists of two steps.
The first step consists of a *change of basis* in which an isometry $\boldsymbol{A} : \mathbb{R}^d \to \mathbb{R}^d$ is introduced, such that

$$\eta = \boldsymbol{A}\xi, \ \boldsymbol{A}\boldsymbol{A}^T = I \tag{1.36}$$

The second step consists of *reduction via projection* where we project the function in a subspace $\mathcal{V}^\eta$ of the new basis function set. The accuracy of approximation is thus clearly dependent on choice of $\boldsymbol{A}$ and the subspace $\mathcal{V}^\eta$. In a non intrusive construction, quadrature based methods can be used for evaluating the inner product (which in this case are expectations w.r.t Gaussian measure). In short, the intuition of this method is to find the directions in which the quantity of interest has low dimensional representation by rotating the coordinates and then to re-project the components of this quantity of interest in the original coordinates.

**Active subspace method [22]:** The essential idea in this method is to detect the direction of strongest variability of $u(\xi)$ by exploiting its gradients and separate the set of variables $\xi_K$ that are aligned along the directions of the strongest

variability from variables $\xi_{K'}$ along which the function is relatively invariant. The next step is to approximate $u(\xi)$ in low dimensional subspace (also called as active subspace) spanned by $\xi_K$.

Briefly, the procedure to find active subspace is as follows. Denoting the gradient of $u(\xi)$ by the column vector $\nabla_\xi u(\xi) = [\frac{\partial u}{\partial \xi_1} \dots \frac{\partial u}{\partial \xi_d}]^T$, the covariance $\mathbf{C}$ of gradient of $u(\xi)$ and its eigen-decomposition is written as

$$\mathbf{C} = E\left[(\nabla_\xi u(\xi))(\nabla_\xi u(\xi))\right] = \mathbf{W}\Upsilon\mathbf{W}^{\mathbf{T}}, \tag{1.37}$$

where $\Upsilon = \text{diag}(\upsilon_1, \dots, \upsilon_d), ; \upsilon_1 \geq \dots \geq \upsilon_d \geq 0$. The first $k < d$ eigenvectors corresponding to $k$ dominant eigenvalues are gathered in $\mathbf{W}_K$ and the rest in $\mathbf{W}_{K'}$. The rotated coordinates $\eta_K \in \mathbb{R}^k$ and $\eta'_K \in \mathbb{R}^{K'}$ are defined by

$$\eta_K = \mathbf{W}_K^T \xi, \ \eta_{K'} = \mathbf{W}_{K'}^T \xi. \tag{1.38}$$

Thus $\eta_K$ are the variables that capture directions along which the function varies strongly and $\boldsymbol{\eta}_{K'}$ are variables along which the function remains relatively flat. The function $u(\xi)$ can then be approximated as

$$u(\xi) \approx u(\xi_K) \tag{1.39}$$

by constructing a response surface w.r.t variables $\xi_K$.
A limitation of this method is that one needs to evaluate the gradient of $u(\xi)$ which may not be available. Also, the covariance matrix $\mathbf{C}$ is constructed using sampling approach such as Monte Carlo which may need a large number of sample evaluations.

## 4.3 Sparsity

Sparse approximation methods rely on the fact that a good approximation of $u(\xi)$ can be obtained by only considering a finite subset of functions in a dictionary $\mathcal{D} = \{\phi_\alpha; \alpha \in \Lambda\}$:

$$u(\xi) \approx u_n(\xi) = \sum_{\alpha \in \Lambda_n} u_\alpha \phi_\alpha(\xi), \ \Lambda_n \subset \Lambda, \ \#\Lambda_n = n. \tag{1.40}$$

Let $\mathcal{M}_n$ be the set of $n$-term decompositions:

$$\mathcal{M}_n = \left\{ \sum_{\alpha \in \Lambda_n} u_\alpha \phi_\alpha(\xi); u_\alpha \in \mathbb{R}, \Lambda_n \subset \Lambda, \ \#\Lambda_n = n\} \right\}. \tag{1.41}$$

The best $n$-term approximation is the solution of

$$\inf_{v \in \mathcal{M}_n} \|u - v\| := \sigma_n(u). \tag{1.42}$$

The existence of an accurate sparse representation is related to the choice of a dictionary which is well adapted to the class of functions considered. For the computation of sparse approximation, we distinguish non adaptive and adaptive methods.

**Non adaptive method**: Let us present these methods using least-squares. The objective here is to obtain a sparse representation of $u(\xi)$ on a basis set $\{\phi_\alpha\} : \alpha \in \Lambda_P\}$ with few evaluations of $u(\xi)$ on $\{y^q\}_{q=1}^Q$ realizations of $\xi$ such that we obtain only $n$ non zero coefficients. The matrix representation of $u = \sum_{\alpha\in\Lambda_P} v_\alpha\phi_\alpha$ for sample realizations $\{y^q\}_{q=1}^Q$ is given by

$$\mathbf{\Phi v} = \mathbf{z}, \tag{1.43}$$

where $\mathbf{z} = (u(y^q))_{q=1}^Q \in \mathbb{R}^Q$, $\mathbf{\Phi} = (\phi_\alpha(y^q)) \in \mathbb{R}^{Q\times P}$ and $\mathbf{v} = (\mathbf{v}_\alpha)_{\alpha\in\Lambda_P} \in \mathbb{R}^P$. The best n-term approximation write

$$\min_{\substack{\mathbf{v}\in\mathbb{R}^P \\ \|v\|_0=n}} \|\mathbf{\Phi v} - \mathbf{z}\|_2^2, \tag{1.44}$$

The above problem is non-convex and we instead solve

$$\min_{\mathbf{v}\in\mathbb{R}^P} \|\mathbf{\Phi v} - \mathbf{z}\|_2^2 + \lambda\|\mathbf{v}\|_1, \tag{1.45}$$

where $\|\mathbf{v}\|_1 = \sum_{\alpha=1}^P |v_\alpha|$. Problem (1.45) is a convex optimization problem for which several methods are readily available. If $u(\xi)$ admits an accurate sparse approximation, then, under some additional conditions, solving (1.45) gives a sparse solution $\mathbf{v}$ [8, 28].

**Adaptive Methods**: The objective of adaptive methods is to select a nested sequence of multi-index sets $\{\Lambda_{n_i}\}_{i=1}^M$ with $\#\Lambda_{n_i} = n_i$ such that $\Lambda_{n_1} \subset \cdots \subset \Lambda_{n_M}$ [18, 20, 21].

Construction of such sets can be chosen from monotone sets i.e. sets with monotone structure

$$\beta \in \Lambda_n \text{ and } \alpha \leq \beta \Rightarrow \alpha \in \Lambda_n, \tag{1.46}$$

where $\alpha \leq \beta \Leftrightarrow \alpha_j \leq \beta_j$ for $\forall j \in \mathbb{N}$. Possible strategies are the bulk search and largest neighbour.

In bulk search approach, a nested sequence of monotone sets $\{\Lambda_{n_i}\}_{i\geq 1}$ is constructed such that

$$\Lambda_{n_i} \subset \Lambda_{n_{i+1}},\ \Lambda_{n_{i+1}} = \Lambda_{n_i} \cup S_i, \tag{1.47}$$

where $S_i \subset \mathcal{N}(\Lambda_{n_i})$ is a subset of the set of neighbours $\mathcal{N}(\Lambda_{n_i})$ of $\Lambda_n$, also called as the margin of $\Lambda_{n_i}$, such that $\Lambda_{n_i} + S_i$ is a monotone set. The margin of a monotone set $\Lambda$ is defined as

$$\mathcal{N}(\Lambda) = \{\alpha \notin \Lambda; \exists j \in \mathbb{N}\ s.t.\ \alpha_j \neq 0 \Rightarrow \alpha - e_j \in \Lambda\}, \tag{1.48}$$

where $e_j$ is the kronecker sequence: $(e_j)_i = \delta_{i,j}$ for $i, j \in \mathbb{N}$. However construction of an optimal subset $S$ in the margin $\mathcal{N}(\Lambda)$ is still an open question.

A modification of this strategy is to add the smallest subset $S$ of the reduced margin $\mathcal{N}_r(\Lambda)$ where

$$\mathcal{N}_r(\Lambda) = \{\alpha \notin \Lambda; \forall j \in \mathbb{N} \; s.t. \; \alpha_j \neq 0 \Rightarrow \alpha - e_j \in \Lambda\}. \tag{1.49}$$

In a non intrusive construction, algorithms using Least square approximation in adaptive monotone sets can be constructed based on posteriori error estimates derived in [19].

## 4.4 Low rank approximations

Another approach based on structured approximation is to assume that $u(\xi)$ can be approximated in a suitable low rank tensor subset [54, 52, 17]. In this approach, $u(\xi)$ is interpreted as a tensor in a tensor product space and low rank approximation of tensors are then exploited to approximate $u(\xi)$. Representation of $u(\xi)$ in a low rank tensor format requires estimation of very few parameters as compared to the dimension of underlying tensor product space and hence one might hope that the curse of dimensionality can be avoided using these structures.

Different notions of rank correspond to different low rank tensor formats. For the approximation of $u(\xi)$ in the simplest low rank format called *canonical tensor format* is given by

$$u(\xi_1, \ldots, \xi_d) \approx \sum_{i=1}^{r} u_i^1(\xi_1) \ldots u_i^d(\xi_d), \tag{1.50}$$

where the rank $r$ of approximation is small. This representation is based on the most basic notion of separated representation of functions [4, 5]. Another notion of rank called *multilinear rank* (based on the definition of minimal subspaces) forms the basis of more general tensor formats. For e.g. the approximation of $u(\xi)$ in *Tucker tensor format* bounded by multilinear rank $r = (r_1, \ldots, r_d) \in \mathbb{N}^d$ is given by

$$u(\xi_1, \ldots, \xi_d) \approx \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} u_{i_1 \ldots i_d} \prod_{k=1}^{d} \phi_{i_k}(\xi_k). \tag{1.51}$$

Note that the curse of dimensionality is not circumvented in this format as the number of terms still depend exponentially on $d$. Recently, other tensor formats called *hierarchical formats* or tree based formats have been proposed that are based on hierarchical separation of dimensions whose parametrization increases only linearly with $d$ [37, 60]. Such tensor formats are discussed in section 3.4.

A particular interesting case for which theoretical results are available is when $u(\xi)$ is a rank one function i.e

$$u(\xi) = u^1(\xi_1) \dots u^d(\xi_d) \tag{1.52}$$

where $u^k : \Xi \to \mathbb{R}$. In [3, 56], authors investigate how well a rank one function can be captured from $N$ well chosen point evaluations. They consider the set of functions

$$F_M = \left\{ u(\xi_1, \dots, \xi_d) = \prod_{k=1}^{d} u^k(\xi_k) : \|u^k\|_\infty \leq 1, \|u^{k^{(s)}}\|_\infty \leq M \right\} \tag{1.53}$$

where $u^{k^{(s)}}$ is $s$-th weak derivative of the univariate function $u^k(\xi_k)$. Under the assumption that $M < 2^s s!$, the number of function evaluations $N$ needed to obtain an approximation error $\epsilon < 1$ is given by $N(\epsilon, d) \backsim O(d^{1+1/s}\epsilon^{-1/s})$. This very particular case illustrates the fact that for functions exhibiting low-rank structures, one may expect to beat the curse of dimensionality.

# Chapter 2

# Low rank tensor formats

*In this chapter, basic notions associated with tensor product spaces are introduced in section 2. The tensor structure of stochastic function spaces leads to the interpretation of a high dimensional stochastic function as a high order tensor. Due to exponential increase in dimensionality of tensor product spaces with the number of parametric inputs, we perform approximation in low rank tensor formats which are introduced in section 3. In section 4, we present low rank approximation algorithms based on singular value decomposition and alternating least-squares. We finally present post-processing of functions represented in low rank tensor formats in section 5.*

# Contents

# 1 Introduction

Approximating a multivariate function $u(\xi)$, $\xi \in \mathbb{R}^d$, requires exploiting some specific structures of the function. A large class of multivariate functions can be well approximated by a sum of $r$ separable functions [6, 7] i.e.

$$u(\xi) \approx \sum_{i=1}^{r} \prod_{k=1}^{d} u_i^k(\xi_k). \tag{2.1}$$

Such an approximation is said to have a *low* separation rank $r \in \mathbb{N}$ if $r$ is small and (2.1) is called a low rank approximation of $u(\xi)$. Such approximations of multivariate functions can be systematically studied in the framework of low rank tensor product approximations. In this chapter, we present the notions associated with tensors and their approximation. The principle references for this chapter are [40, 46, 38] and chapter 2 of [36].

# 2 Tensor Spaces

In this section, we recall basic definitions related to tensors and tensor product spaces. We then show that a high dimensional stochastic function can be seen as an element of a tensor product space.

## 2.1 Tensor product space

Let $V^k$, $k \in D = \{1, \ldots, d\}, d \in \mathbb{N}$, be $d$ vector spaces equipped with norms $\|\cdot\|_k$. Let us first define elementary tensors.

**Definition 2.1.** *(Elementary tensor) We call a tensor $v$ an elementary tensor if there exists $v^k \in V^k, 1 \leq k \leq d$ such that*

$$v = v^1 \otimes \ldots \otimes v^d. \tag{2.2}$$

With this definition, we define the algebraic tensor product space as follows.

**Definition 2.2.** *(Algebraic tensor product space) Algebraic tensor product space $V = \otimes_{k=1}^{d} V^k$ is the linear span of elementary tensors i.e.*

$$V = \bigotimes_{k=1}^{d} V^k = span\left\{ v = \bigotimes_{k=1}^{d} v^k : v^k \in V^k, 1 \leq k \leq d \right\}. \tag{2.3}$$

Thus, an element $v \in V$ can be written as finite linear combinations of elementary tensors, i.e.

$$v = \sum_{i=1}^{m} v_i^1 \otimes \cdots \otimes v_i^d, \tag{2.4}$$

for some $m \in \mathbb{N}$ and some vectors $v_i^k \in V^k, 1 \leq i \leq m, 1 \leq k \leq d$.
A *tensor Banach space* $V_{\|\cdot\|}$ equipped with a norm $\|\cdot\|$ is defined as the completion of an algebraic tensor space $V$. If $V^k$ are Hilbert spaces with a norm $\|\cdot\|_k$, associated with a scalar product $\langle\cdot,\cdot\rangle_k$, a natural (canonical) inner product can be defined for $v, w \in V$ by

$$\langle v, w\rangle = \prod_{k=1}^{d} \langle v^k, w^k\rangle, \tag{2.5}$$

and extended by linearity for arbitrary tensors. The norm associated with this scalar product is called the *canonical norm*. The resulting space $V_{\|\cdot\|}$ is a tensor Hilbert space.
Having defined a tensor Hilbert space, in the following we will show that a multivariate stochastic function $u(\xi)$ can be interpreted as a tensor.

## 2.2 Tensor structure of stochastic function spaces

Let us consider that the function $u(\xi)$ is a function of $d$ independent random variables $\xi = \{\xi_k\}_{k=1}^d$. We denote by $(\Xi, \mathcal{B}, P_\xi)$ the probability space induced by $\xi$, where $\Xi \subset \mathbb{R}^d$ and where $P_\xi$ is the probability law of $\xi$. We denote by $(\Xi_k, \mathcal{B}_k, P_{\xi_k})$ the probability space associated with $\xi_k$, where $P_{\xi_k}$ is the probability law of $\xi_k$. The probability space $(\Xi, \mathcal{B}, P_\xi)$ associated with $\xi = (\xi_1, \ldots \xi_d)$ has a product structure with $\Xi = \times_{k=1}^d \Xi_k$, $\mathcal{B} = \otimes_{k=1}^d \mathcal{B}_k$ and $P_\xi = \otimes_{k=1}^d P_{\xi_k}$. We denote by $L^2_{P_\xi}(\Xi)$ the Hilbert space of second order random variables defined on $(\Xi, \mathcal{B}, P_\xi)$, defined by

$$L^2_{P_\xi}(\Xi) = \left\{ u : y \in \Xi \mapsto u(y) \in \mathbb{R}; \int_\Xi u(y)^2 P_\xi(dy) < \infty \right\}.$$

$L^2_{P_\xi}(\Xi)$ is a tensor Hilbert space with the following structure:

$$L^2_{P_\xi}(\Xi) = L^2_{P_{\xi_1}}(\Xi_1) \otimes \ldots \otimes L^2_{P_{\xi_d}}(\Xi_d). \tag{2.6}$$

We can introduce approximation spaces $\mathcal{S}^k_{n_k} \subset L^2_{P_{\xi_k}}(\Xi_k)$ with orthonormal basis $\{\phi^k_j\}_{j=1}^{n_k}$, such that

$$\mathcal{S}^k_{n_k} = \left\{ v^k(y_k) = \sum_{j=1}^{n_k} v^k_j \phi^k_j(y_k); v^k_j \in \mathbb{R} \right\} = \left\{ v^k(y_k) = \boldsymbol{\phi}^k(y_k)^T \mathbf{v}^k; \mathbf{v}^k \in \mathbb{R}^{n_k} \right\},$$

where $\mathbf{v}^k$ denotes the vector of coefficients of $v^k$ and where $\boldsymbol{\phi}^k = (\phi^k_1, \ldots, \phi^k_{n_k})^T$ denotes the vector of basis functions. An approximation space $\mathcal{S}_n \subset L^2_{P_{\boldsymbol{\xi}}}(\Xi)$ is then obtained by tensorization of approximation spaces $\mathcal{S}^k_{n_k}$:

$$\mathcal{S}_n = \mathcal{S}^1_{n_1} \otimes \ldots \otimes \mathcal{S}^k_{n_k}.$$

An element $v \in \mathcal{S}_n$ can be written

$$v = \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} v_{i_1 \dots i_d} \bigotimes_{k=1}^{d} \phi_{i_k}^k = \sum_{i \in I_n} v_i \phi_i \tag{2.7}$$

where $I_n = \times_{k=1}^d \{1 \dots n_k\}$ and $\phi_i(y) = (\phi_{i_1}^1 \otimes \dots \otimes \phi_{i_d}^d)(y_1, \dots, y_d) = \phi_{i_1}^1(y_1) \dots \phi_{i_d}^d(y_d)$. An element $v = \sum_i v_i \phi_i \in \mathcal{S}_n$ can thus be identified with the algebraic tensor $\mathbf{v} \in \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$ such that $(\mathbf{v})_i = v_i$. Denoting $\boldsymbol{\phi}(y) = \boldsymbol{\phi}^1(y_1) \otimes \dots \otimes \boldsymbol{\phi}^d(y_d) \in \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$, we have the identification $\mathcal{S}_n \simeq \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$ with

$$\mathcal{S}_n = \{v(y) = \langle \boldsymbol{\phi}(y), \mathbf{v} \rangle ; \mathbf{v} \in \mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}\},$$

where $\langle \cdot, \cdot \rangle$ denotes the canonical inner product in $\mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$.

**Remark 2.1 :**
*In the case where the random variables are not independent, $L^2_{P_\xi}(\Xi)$ is no longer a tensor product space. However, we can define an orthogonal approximation basis $(\phi_i)_{i \in I_n}$ of $L^2_{P_\xi}(\Xi)$ as follows [72]*

$$\phi_i(y) = \prod_{k=1}^{d} \phi_{i_k}^k(y_k) \sqrt{\frac{p_{\xi_k}(y_k)}{p_\xi(y)}}, \tag{2.8}$$

*where $p_\xi$ and $p_{\xi_k}$ are the probability density functions of $\xi$ and $\xi_k$ respectively. We can define the approximation space $\mathcal{S}_n = span\{\phi_i ; i \in I_n\}$ which does not have the tensor product structure but is isomorphic to $\mathbb{R}^{n_1} \otimes \dots \otimes \mathbb{R}^{n_d}$.*

The dimension of approximation space $\mathcal{S}_n$ grows exponentially with stochastic dimension $d$, since $\dim(\mathcal{S}_n) = \prod_{k=1}^d n_k$. Thus one faces complexity issues when approximating a function $u \in \mathcal{S}_n$ when $d$ is large. To reduce the complexity, we perform approximation in low rank tensor subsets of $\mathcal{S}_n$ that aims at finding a representation of the form (2.7) with a low dimensional representation of the coefficients $\mathbf{v}$. In the following, we will omit the indices $n$ and $n_k$ in $\mathcal{S}_n$ and $\mathcal{S}_{n_k}^k$ for simplicity of notations.

# 3 Low rank tensor formats

In this section, we briefly recall main low rank tensor formats. These formats are associated with different notions of rank of a tensor.
We can interpret these low rank tensor formats as multi-linear maps $F_{\mathcal{M}} : \mathcal{P}^1 \times \dots \times \mathcal{P}^m \to \mathcal{M}$ where $\mathcal{M}$ is a low rank tensor subsets and $\mathcal{P}^i, 1 \leq i \leq m$, are vector spaces of real parameters. For $v \in \mathcal{M}$, there exists $(\mathbf{v}^1, \dots, \mathbf{v}^m) \in \mathcal{P}^1 \times \dots \times \mathcal{P}^m$ such that

$$v = F_{\mathcal{M}}(\mathbf{v}^1, \dots, \mathbf{v}^m). \tag{2.9}$$

In the following, we will specify spaces $\mathcal{P}^i, 1 \leq i \leq m$, and $F_{\mathcal{M}}$ for different low rank tensor formats.

## 3.1 Canonical tensor format

The canonical rank of a tensor $v \in \mathcal{S} = \mathcal{S}^1 \otimes \ldots \otimes \mathcal{S}^d$ is the smallest integer $r \in \mathbb{N}$ such that

$$v = \sum_{i=1}^{r} \otimes_{k=1}^{d} v_i^k, \tag{2.10}$$

for $v_i^k \in \mathcal{S}^k,\ 1 \leq i \leq r,\ 1 \leq k \leq d$. The set of (elementary) rank-one tensors $\mathcal{R}_1$ is defined by

$$\mathcal{R}_1 = \left\{ v = \otimes_{k=1}^{d} v^k;\ \ v^k \in \mathcal{S}^k \right\}.$$

The set of tensors with canonical rank at most $r$ is defined as

$$\mathcal{R}_r = \left\{ v = \sum_{i=1}^{r} \otimes_{k=1}^{d} v_i^k;\ \ v_i^k \in \mathcal{S}^k \right\}.$$

The set $\mathcal{R}_r$ is not closed for $d \geq 3$ and $r > 1$ [24]. This property is important because the existence of best approximation is only guaranteed for closed sets. To avoid this difficulty, we can define the set of bounded canonical tensors with bounded factors

$$\mathcal{R}_r^c = \left\{ v = \sum_{i=1}^{r} \otimes_{k=1}^{d} v_i^k;\ \ v_i^k \in \mathcal{S}^k, \| \otimes_{k=1}^{d} v_i^k \| \leq c \right\},$$

With $c > 0$, $\mathcal{R}_r^c$ is closed whatever $r$ and $d$ (see Lemma 4.2 [32]).

**Parametrization:** For a given choice of basis of $\mathcal{S}^k, 1 \leq k \leq d$, elements of $v \in \mathcal{R}_r$ can be written $v = F_{\mathcal{R}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$, where $F_{\mathcal{R}_r}$ is a multilinear map parametrising the subset $\mathcal{R}_r$ and $\mathbf{v}^k \in \mathcal{P}^k = \mathbb{R}^{n_k \times r}, 1 \leq k \leq d$, such that $\mathbf{v}^k = (\mathbf{v}_1^k \ldots \mathbf{v}_r^k)$ are the coefficients of the vectors $v_i^k \in \mathcal{S}^k, 1 \leq i \leq r$ on the chosen basis of $\mathcal{S}^k$. The total number of real parameters is therefore $r \sum_{k=1}^{d} n_k$.

## 3.2 Minimal subspaces and $t$-rank

The definition of subspace based tensor formats is linked to the concept of minimal subspaces and $t$-ranks. For a fixed $v \in \otimes_{k=1}^{d} \mathcal{S}^k$, the minimal subspace $U_{min}^k(v)$ is the smallest subspace $U^k \subset \mathcal{S}^k$ such that

$$v \in U^k \otimes \mathcal{S}^{[k]}, \tag{2.11}$$

with $\mathcal{S}^{[k]} = \otimes_{l \neq k} \mathcal{S}^l$ (here we use a permutation of dimensions for ease of notations). For $t \subset D$ and $t^c = D \setminus t$ such that $t$ and $t^c$ are non empty, the minimal subspace

$U^t_{min}(v)$ is defined as the smallest subspace $U^t$ of $\mathcal{S}^t = \otimes_{k\in t}\mathcal{S}^k$ such that $v \in U^t \otimes \mathcal{S}^{t^c}$. The $t$-rank of $v$ is then given by

$$\text{rank}_t(v) = \dim(U^t_{min}(v)) \tag{2.12}$$

We can define a linear map $\mathcal{M}_t : \otimes_{k=1}^d \mathcal{S}^k \to \mathcal{S}^t \otimes \mathcal{S}^{t^c}$, called matricisation operator, defined for elementary tensors by

$$\mathcal{M}_t\left(\bigotimes_{k\in D} v^k\right) = \left(\bigotimes_{k\in t} v^k\right) \otimes \left(\bigotimes_{k\in t^c} v^k\right), \tag{2.13}$$

and extended by linearity to $\underset{k\in D}{\otimes} \mathcal{S}^k$. $\mathcal{M}_t(v)$ is called as *matricisation* or *unfolding* of $v$. The $t$-rank of $v$ is given by the rank of order-two tensor $\mathcal{M}_t(v)$ [41, 37]. Based on these definitions, we can now define low rank subsets by considering $t$-rank for a collection of subsets $t$.

## 3.3 Tucker tensor format

The Tucker (or multilinear) rank of $v \in \mathcal{S}$ is defined as the rank tuple $r = (r_1, \ldots, r_d)$ where $r_k = \text{rank}_k(v),\ k \in D$. The set of *Tucker Tensors* of multilinear rank bounded by $r$ is given by

$$\mathcal{T}_r = \{v \in \mathcal{S}; \text{rank}_k(v) = \dim(U^k_{min}) \leq r_k,\ k \in D\}. \tag{2.14}$$

An element $v \in \mathcal{T}_r$ can be written under the form

$$v = \sum_{i_1=1}^{r_1} \cdots \sum_{i_d=1}^{r_d} \alpha_{i_1 \ldots i_d} \otimes_{k=1}^d v^k_{i_k}, \tag{2.15}$$

where $\alpha \in \mathbb{R}^{r_1 \times \ldots \times r_d}$ is called the *core tensor* and $v^k_{i_k} \in \mathcal{S}^k,\ k \in D;\ 1 \leq i_k \leq r_k$. The set $\mathcal{T}_r$ with multilinear rank bounded by $r$ is closed.

**Parametrization:** An element $v \in \mathcal{T}_r$ can be written as $v = F_{\mathcal{T}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^{d+1})$, where $F_{\mathcal{T}_r}$ parametrizes the subset $\mathcal{T}_r$ and $\mathbf{v}^k = (\mathbf{v}^k_1 \ldots \mathbf{v}^k_{r_k}) \in \mathcal{P}^k = \mathbb{R}^{n_k \times r_k}, 1 \leq k \leq d$, parametrising the vectors $v^k_{i_k} \in \mathcal{S}^k, 1 \leq i_k \leq r_k$, and $\mathbf{v}^{d+1} = \alpha \in \mathbb{R}^{r_1 \times \ldots \times r_d}$. There are therefore $\prod_{k=1}^d r_k + \sum_{k=1}^d r_k n_k$ real parameters. Note that the number of parameters still increases exponentially with $d$ (in the core tensor) and hence this format suffers from curse of dimensionality.

## 3.4 Tree based formats

Tree based formats are based on a more general notion of rank for a group of dimensions associated with a dimension partition tree. Let $D = \{1, \ldots, d\}$ and $T$ be a dimension partition tree on $D$, which is a subset of $2^D$ such that every vertex

$t \in T$ are non empty subsets of $D$. Let us denote $L(T)$ as the leaves of $T$ and $I(T) = T \setminus L(T)$, so that $L(T) = \{\{k\} : k \in D\}$. An element $t \in L(T)$ is defined such that $\#t = 1$. Let $S(t)$ denote sons of $t \in T$. The depth of the tree is defined as $p = \lceil \log_2(d) \rceil := \min\{i \in \mathbb{N} | i \geq \log_2(d)\}$. The level $l$ of the tree is defined as the set of all nodes having a distance of exactly $l$ to the root D. We denote the level $l$ of the tree as $T^l := \{t \in T | level(t) = l\}$. In the following, we will consider two particular tree based tensor formats.

### 3.4.1 Hierarchical tensor format

Hierarchical tensor format [41] is associated with a binary tree $T$, such that for all $t \in I(t)$, $\#S(t) = 2$. An example of a binary dimension tree for $D = \{1, 2, 3, 4\}$ is illustrated in figure 2.1.
The Hierarchical tensor rank of a tensor $v$ is a tuple $(\text{rank}_t(v))_{t \in T} \in \mathbb{N}^{\#T}$. For a rank tuple $r = (r_t)_{t \in T}$, the subset of Hierarchical tucker tensors with $T-$rank bounded by $r$ is defined as

$$\mathcal{H}_r^T = \{v \in \mathcal{S} : \text{rank}_t(v) = \dim(U_{min}^t(v)) \leq r_t, t \in T\}. \tag{2.16}$$

The rank $r$ is admissible if there is at least one element in $v \in \mathcal{H}_r^T \setminus \{0\}$. For $t \in T$, we denote by $(v_i^t)_{1 \leq i \leq r_t}$ a basis of $U_{min}^t(v)$. For $t \in I(t)$, with $S(t) = \{t_1, t_2\}$, we can write

$$v_i^t = \sum_{\substack{1 \leq j \leq r_{t_1} \\ 1 \leq k \leq r_{t_2}}} \alpha_{ijk}^t v_j^{t_1} \otimes v_k^{t_2}. \tag{2.17}$$

for $1 \leq i \leq r_t$. The tensor $\alpha^t \in \mathbb{R}^{r_t \times r_{t_1} \times r_{t_2}}, t \in I(t)$, are called the *transfer tensors*. With $r_D = 1$, an element $v \in \mathcal{H}_r^T$ is represented as

$$v = \sum_{i=1}^{r_{D_1}} \sum_{j=1}^{r_{D_2}} \alpha_{ij}^D v_i^{D_1} \otimes v_j^{D_2}, \tag{2.18}$$

where $S(D) = \{D_1, D_2\}$. A tensor $v \in \mathcal{H}_r^T$ is completely determined by the transfer tensors $(\alpha^t)_{t \in I(T)}$ and the vectors $(v_i^k)_{k \in L(T), 1 \leq i \leq r_k}$.

**Parametrisation:** An element $v \in \mathcal{H}_r^T$ can be written as $v = F_{\mathcal{H}_r^T}(\mathbf{v}^1, \ldots, \mathbf{v}^m)$, $m = \#I(T) + \#L(T)$, where $F_{\mathcal{H}_r^T}$ parametrizes the subset $\mathcal{H}_r^T$. Here $\mathbf{v}^k = (\mathbf{v}_1^k \ldots \mathbf{v}_{r_k}^k) \in \mathcal{P}^k = \mathbb{R}^{n_k \times r_k}, 1 \leq k \leq d$, parametrizes the leaf node vectors $\{v_{i_k}^k\}_{i_k=1}^{r_k} \in \mathcal{S}^k$ and $\{\mathbf{v}^{d+1}, \ldots, \mathbf{v}^m\} = \{\alpha^t : t \in I(T)\}$. There are therefore $\sum_{k=1}^d r_k n_k + \sum_{t \in I(T)} r_t r_{t_1} r_{t_2}$ real parameters.

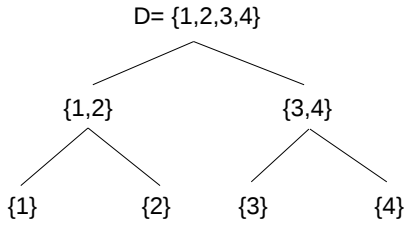



**Figure 2.1:** Dimension partition tree ($d = 4$)

### 3.4.2 Tensor Train tensor format

The Tensor Train format [60, 61, 59] is another particular case of tree based tensors. The dimension partition tree of a Tensor Train format is characterised by $S(t) = (t_1, t_2)$ such that $t_1 = \{k\}$ and $t_2 = \{k+1, \ldots, d\}$ for $1 \leq k \leq d-1$. An example of dimension partition tree for Tensor Train format is shown in figure 2.2.

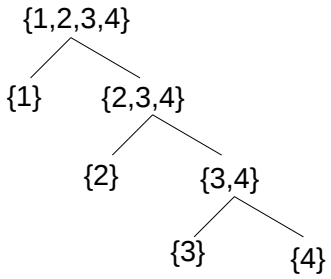



**Figure 2.2:** Dimension partition tree for Tensor Train format ($d = 4$)

The tensor train rank $r = (r_1 \ldots, r_{d-1})$ is a tuple with $r_k = \text{rank}_{t_k}(v)$ where $t_k = \{k, \ldots, d\}$. The set of Tensor Train tensors of rank bounded by $r$ is given by

$$\mathcal{TT}_r = \left\{ v \in \mathcal{S};\ \text{rank}_t(v) = \dim(U^t_{min}(v)) \leq r, t \in I(T) \right\}. \tag{2.19}$$

An equivalent representation of an element $v \in \mathcal{TT}_r$ can be given under the form

$$v = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{d-1}=1}^{r_{d-1}} \bigotimes_{k=1}^{d} v^k_{i_{k-1} i_k};\ v^k_{i_{k-1} i_k} \in \mathcal{S}^k, \tag{2.20}$$

with $i_0 = i_d = 1$. A more compact representation is given by

$$v(\xi) = G^1(\xi_1) \cdots G^d(\xi_d),\ G^k \in (\mathcal{S}^k)^{r_{k-1} \times r_k}, \tag{2.21}$$

where $G^k_{i_{k-1}i_k} = v^k_{i_{k-1}i_k}$. The set $\mathcal{TT}_r$ with tensor train rank bounded by $r = (r_1 \dots, r_{d-1})$ is closed.

**Parametrization:** An element $v \in \mathcal{TT}_r$ can be written as $v = F_{\mathcal{TT}_r}\left(\mathbf{v}^1, \dots, \mathbf{v}^d\right)$ where $F_{\mathcal{TT}_r}$ parametrizes the subset $\mathcal{TT}_r$. Here $\mathbf{v}^k \in \mathcal{P}^k = \mathbb{R}^{r_{k-1} \times r_k \times n_k}, 1 \leq k \leq d$, parametrizes $v^k_{i_{k-1}i_k} \in \mathcal{S}^k, 1 \leq i_{k-1} \leq r_{k-1}, 1 \leq i_k \leq r_k$. There are therefore $\sum_{k=1}^d r_{k-1} r_k n_k$ real parameters.

# 4 Tensor approximation

In this section, we present algorithms for tensor approximations. The idea is to search for an approximation of $v \in \mathcal{S}$ in a particular tensor subset $\mathcal{M}$ (where $\mathcal{M}$ can be $\mathcal{R}_r$, $\mathcal{T}_r$, $\mathcal{H}^T_r$ or $\mathcal{TT}_r$). A best approximation $u_r$ of $u$ in $\mathcal{M}$ (if it exists) is such that

$$\|u - u_r\| = \min_{v \in \mathcal{M}} \|u - v\|. \tag{2.22}$$

## 4.1 Singular value decomposition

### 4.1.1 Singular value decomposition ($d = 2$)

We consider the case when $d = 2$, thus $\mathcal{S} = \mathcal{S}^1 \otimes \mathcal{S}^2$. We denote $n = \min(n_1, n_2)$. Let $u \in \mathcal{S}$, then there exists a decreasing sequence of positive numbers $\sigma = (\sigma_i)_{i=1}^n$ and two orthonormal systems $(v^1_i)_{1 \leq i \leq n} \subset \mathcal{S}^1$ and $(v^2_i)_{1 \leq i \leq n} \subset \mathcal{S}^2$ such that

$$u = \sum_{i=1}^n \sigma_i v^1_i \otimes v^2_i. \tag{2.23}$$

The expression (2.23) is called the *singular value decomposition* of $u$. The $(\sigma_i)_{1 \leq i \leq n} \in (\mathbb{R}^+)^n$ are called the singular values, $(v^1_i)_{1 \leq i \leq n} \in (\mathcal{S}^1_{n_1})^n$ and $(v^2_i)_{1 \leq i \leq n} \in (\mathcal{S}^2_{n_2})^n$ are called the left and right singular vectors of $u$. If $\mathcal{S}$ is endowed with a canonical norm, a best rank-$r$ approximation of $u$, solution of (2.22), is given by

$$u_r = \sum_{i=1}^r \sigma_i v^1_i \otimes v^2_i \tag{2.24}$$

with $\|u - u_r\|^2 = \sum_{i=r+1}^n \sigma_i^2$. The $r$-dimensional subspaces

$$\mathcal{S}^1_r = \operatorname{span}\{v^1_i\}_{i=1}^r \text{ and } \mathcal{S}^2_r = \operatorname{span}\{v^2_i\}_{i=1}^r \tag{2.25}$$

are respectively the left and right dominant singular spaces of $v$. Thus, the SVD defines increasing sequences of optimal subspaces $\{\mathcal{S}^1_r\}_{r \geq 1}$ and $\{\mathcal{S}^2_r\}_{r \geq 1}$ such that,

$$\mathcal{S}^1_r \subset \mathcal{S}^1_{r+1} \text{ and } \mathcal{S}^2_r \subset \mathcal{S}^2_{r+1}.$$

### 4.1.2 Higher order SVD for $d > 2$

In this section, we consider the problem of the low rank approximation of higher order tensors based on the generalization of the SVD.

**Higher-Order Singular Value Decomposition (HOSVD).** This method, introduced in [23] for Tucker format, searches an approximation of a tensor $v \in \mathcal{S}_n$ in the subset $\mathcal{T}_r$. It relies on the use of the SVD for order-2 tensors obtained by matricisation of $v$ defined in section 3.2. In this algorithm, for $1 \leq k \leq d$, we determine the optimal $r_k$-dimensional space $\mathcal{W}^k_{r_k}$ spanned by the left singular vectors $(v^k_{i_k})_{1\leq i_k \leq r_k}$ of $\mathcal{M}_{\{k\}}(u)$. Then, the HOSVD of rank $r = (r_1, \ldots, r_d)$ of $u$ is defined as the best approximation of $u$ in $\mathcal{W}^1_{r_1} \otimes \cdots \otimes \mathcal{W}^d_{r_d}$, written as

$$u_r = \sum_{i_1=1}^{r_1} \cdots \sum_{r_d=1}^{r_d} \alpha_{i_1 \ldots i_d} \bigotimes_k v^k_{i_k}, \tag{2.26}$$

with a core tensor $\alpha$ given by

$$\alpha_{i_1 \ldots i_d} = \left\langle u, \bigotimes_k v^k_{i_k} \right\rangle \tag{2.27}$$

Algorithm 1 gives the HOSVD for the Tucker format.

**Algorithm 1** HOSVD

**Input:** $u \in \mathcal{S}$.
**Output:** $u_r \in \mathcal{T}_r$.
1: **for** $k = 1, \ldots, d$ **do**
2: Compute $(v^k_i)_{1\leq i \leq r_k}$ as left singular vectors of $\mathcal{M}_{\{k\}}(u)$;
3: **end for**
4: Update $\alpha$ such that $\alpha_{i_1 \ldots i_d} = \left\langle u, \bigotimes_k v^k_{i_k} \right\rangle$;

We have that $u_r$ is a quasi-optimal approximation of $u$ in $\mathcal{T}_r$, such that

$$\|u - u_r\| \leq \sqrt{d} \min_{v \in \mathcal{T}_r} \|u - v\| \tag{2.28}$$

Various alternatives to improve the approximation provided by HOSVD have been developed including recent developments based on Newton type methods [31, 45], Jacobi algorithms [44] and modifications of HOSVD [77].

**The Hierarchical SVD (SVD)** is a method introduced in [37] and is a generalization of HOSVD for Hierarchical Tensor formats. Here, we consider the $t$-matricisation $\mathcal{M}_t(u)$ of $u$ and construct the optimal subspace $\mathcal{W}^t_{r_t} \subset \mathcal{S}^t$ spanned

**Algorithm 2** HSVD

**Input:** $u \in \mathcal{S}$.
**Output:** $v \in \mathcal{H}_r^T$.
1: **for** $t \in L(T)$ **do**
2: Calculate $(v_i^t)_{1\leq i\leq r_t}$ as left singular vectors of $u^{\{k\}}$;
3: **end for**
4: **for** $l = p-1, \ldots, 0$ **do**
5: **for** $t \in I(T)$ on level $l$ **do**
6: Calculate $(v_i^t)_{1\leq i\leq r_t}$ as left singular vectors of $\mathcal{M}_t(u)$;
7: Noting $t = t_1 \cup t_2$, calculate $\alpha^t$ such that $\alpha_{ijk}^t = \left\langle v_k^t, v_i^{t_1} \otimes v_j^{t_2} \right\rangle$;
8: **end for**
9: **end for**
10: Noting $D = t_1 \cup t_2$, calculate $\alpha^D$ such that $\alpha_{ij}^D = \langle u, v^{t_1} \otimes v^{t_2} \rangle$

by dominant left singular vectors $(v_i^t)_{1\leq i\leq r_t}$ of $\mathcal{M}_t(u)$. Then, the HSVD is obtained by applying successively projections on the subspace. A basic version of HSVD is given by Algorithm 2. We have that $u_r$ is a quasi-optimal approximation of $u$ in $\mathcal{H}_r^T$, such that

$$\|u - u_r\| \leq \sqrt{2d-3} \min_{w \in \mathcal{H}_r^T} \|u - w\|. \tag{2.29}$$

## 4.2 Alternating Least-Squares Algorithm

With a parametrization $v = F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m)$ of a low rank tensor formats, one can use classical optimization techniques (conjugate gradient, line search etc) for approximation of tensors in low rank subsets. One of the widely used algorithm, Alternating Least-Squares (ALS), is presented below.

The best approximation problem in $\mathcal{M}$ can be written

$$\begin{aligned} &\text{Find } (\mathbf{v}^1, \ldots, \mathbf{v}^m) \in \mathcal{P}_{\mathcal{M}}^1 \times \cdots \times \mathcal{P}_{\mathcal{M}}^m \text{ such that} \\ &\|u - F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m)\|^2 = \min_{\mathbf{p}^1, \ldots, \mathbf{p}^m \in \mathcal{P}_{\mathcal{M}}^1 \times \cdots \times \mathcal{P}_{\mathcal{M}}^1} \|u - F_{\mathcal{M}}(\mathbf{p}^1, \ldots, \mathbf{p}^m)\|^2. \end{aligned} \tag{2.30}$$

The alternating minimization algorithm consists in successively minimizing over each variable $\mathbf{v}^i \in \mathcal{P}_{\mathcal{M}}^i$, as shown in algorithm 3. For $F_{\mathcal{M}}$ a multilinear map, $F_{\mathcal{M}}$ is linear in each variable $v^k$, and therefore the minimization in $v^k$ is a simple quadratic optimization problem. The ALS is easy to implement but may converge slowly to a stationary point. The limit may also depend on the initial guess (Step 1).

### 4.2.1 Alternating Least Squares in low rank tensor formats

For canonical tensor format, the first question that we ask is: does a best approximation exist in $\mathcal{R}_r$. Unfortunately, it is not necessarily the case for $d > 2$ and $r > 1$

**Algorithm 3** Alternating Least Squares in $\mathcal{P}^1_{\mathcal{M}} \times \dots \mathcal{P}^m_{\mathcal{M}}$

**Input:** $u \in \mathcal{S}_n$.
**Output:** $v \in \mathcal{M}$.

```
Initialize (v^1, ..., v^m) randomly;
v = F_M(v^1, ..., v^m);
while v has not converged do
    for λ = 1, ..., m do
        v^λ = argmin_{w ∈ P^λ_M} ||u − F_M(v^1, ..., w, ..., v^m)||;
        v = F_M(v^1, ..., v^m);
    end for
end while
return v = F_M(v^1, ..., v^m);
```

since $\mathcal{R}_r$ is not closed [24]. However, ALS has proved useful for a large class of numerical examples.

For Tucker tensor format, the ALS consists of minimizing over the parameters $\mathbf{v}^k, 1 \leq k \leq d$, followed by the core tensor $\alpha$. Approximating in Hierarchical Tucker subset consists of minimizing alternately over the parameters associated to different nodes of the tree. This leads to solving a linear system of equations for each node under the form

$$A^t \alpha^t = b^t, \ \forall t \in I(T) \text{ or } A^t v^t = b^t, \ \forall t \in L(T).$$

For the construction of operators $(A^t)_{t \in T}$ and $(b^t)_{t \in T}$, we refer to [47]. In tensor train format, the ALS consists in minimizing over the parameters $\mathbf{v}^k \in \mathbb{R}^{r_{k-1} \times r_k \times n_k}, k \in D$.

### 4.2.2 Modified Alternating Least Squares

Here presented for Tensor Train format, Modified Alternating Least Squares (also known as Density Matrix Renormalization Group) is similar to ALS except that it involves simultaneous minimization on parameters by performing contraction along adjacent cores $(\mathbf{v}^k, \mathbf{v}^{k+1}), 1 \leq k \leq d-1$. Here, in each iteration, the parameters $\mathbf{v}^k \in \mathcal{P}^k_{\mathcal{M}}$ and $\mathbf{v}^{k+1} \in \mathcal{P}_{\mathcal{M}}$ are replaced by the parameter $\mathbf{w}^* \in \mathcal{P}^*_{\mathcal{M}} = \mathbb{R}^{r_{k-1} \times n_k \times r_{k+1} \times n_{k+1}}$ parametrizing $(\mathcal{S}^k \otimes \mathcal{S}^{k+1})^{r_{k-1} \times r_{k+1}}$. We then minimize with respect to $\mathbf{w}^*$ and perform a truncated SVD of the matricization $\mathbf{W} \in \mathbb{R}^{r_{k-1} n_k \times r_{k+1} n_{k+1}}$ with rank $r^*$ to obtain the new parameters $\mathbf{v}^k \in \mathcal{P}^k_{\mathcal{M}} = \mathbb{R}^{r_{k-1} \times r^* \times n_k}$ and $\mathbf{v}^{k+1} \in \mathcal{P}^{k+1}_{\mathcal{M}} = \mathbb{R}^{r^* \times r_{k+1} \times n_{k+1}}$. The rank $r^*$ is selected to achieve a prescribed accuracy. In a sampling based setting, it can be selected through cross-validation techniques (see chapter 5).

**Remark 2.2 :**
*For hierarchical tensor format, this algorithm can be extended by performing contraction for pairs of nodes and minimizing with respect to contracted parameters followed by an SVD [76].*

# 5 Post processing in low rank tensor formats

Let $v(\xi)$ denote a low rank approximation of a multivariate stochastic function $u(\xi) \in \mathcal{S}_n$ given by

$$u(\xi) \approx v(\xi) = \sum_{i \in I_n} \alpha_i \left( \bigotimes_k v_{i_k}^k \right)(\xi) \tag{2.31}$$

with a possibly structured tensor $\alpha$.

## 5.1 Estimation of moments

Solution statistics of integral form such as mean and variance can be computed analytically. Since low rank approximations are based on separated representation, moments of $u(\xi)$ can be obtained using 1-D integrations. In particular, the mean is given by

$$E(v(\xi)) = \sum_i \alpha_i E\left( \prod_k v_{i_k}^k(\xi_k) \right) = \sum_i \alpha_i \prod_k E(v_{i_k}^k(\xi_k)). \tag{2.32}$$

For $\mathcal{S}_{n_k}^k = \text{span}\{\phi_j^k(\xi_k);\ 1 \le j \le n_k\}$, we have

$$E(v_i^k(\xi_k)) = E(\sum_{j=1}^{n_k} v_{i,j}^k \phi_j^k(\xi_k)) = \sum_{j=1}^{n_k} v_{i,j}^k \gamma_j^k, \tag{2.33}$$

with $\gamma_j^k = E(\phi_j^k(\xi_k))$ usually known analytically for standard bases (e.g. polynomial) and probability measures. Similarly, when considering orthonormal bases $\{\phi_j^k\}_{j=1}^{n_k}$,

$$E(v(\xi)^2) = \sum_i \sum_{i'} \alpha_i \alpha_{i'} \prod_k E(v_{i_k}^k(\xi_k) v_{i'_k}^k(\xi_k)) = \sum_i \sum_{i'} \alpha_i \alpha_{i'} \prod_{k=1}^d \left( \sum_{j=1}^{n_k} v_{i_k,j}^k v_{i'_k,j}^k \right). \tag{2.34}$$

Alternatively, we may draw random realizations of $v(\xi)$ and obtain Monte Carlo estimates of the statistics of $u(\xi)$ at negligible cost.

## 5.2 Estimation of global sensitivity indices

Sobol indices [70, 68, 69] are good estimates of the impact of some random variables on the variability of $u(\xi)$. For a single random variable $\xi_k$, the first order Sobol index $S_k$ is defined by

$$S_k = \frac{V(E(u(\xi)|\xi_k))}{V(u)}, \tag{2.35}$$

where $E(u(\xi)|\xi_k)$ is the conditional expectation of $u(\xi)$ w.r.t. $\xi_k$. For approximations of the form (2.31), we have

$$E(u(\xi)|\xi_k) = \sum_{i_k=1}^{r_k} \beta_{i_k}^k v_i^k(\xi_k) \text{ where } \beta_{i_k}^k = \sum_{\substack{1\le i_l\le r_l \\ l\neq k}} \alpha_i \prod_{l\neq k} E(v_{i_l}^l(\xi_l)). \tag{2.36}$$

Using (2.36), we can analytically estimate first order Sobol sensitivity indices.
For a group of random variables $\xi_K$, $K \subset D$, we define the total sensitivity index $S_K^T$ and closed index $S_K^C$ respectively by

$$S_K^T = \sum_{J\cap K\neq\emptyset} S_J, \quad S_K^C = \sum_{J\subset K} S_J. \tag{2.37}$$

Denoting $K^c = D \setminus K$, we have the property

$$S_{K^c}^C + S_K^T = 1. \tag{2.38}$$

These indices can be expressed in terms of conditional expectations and variances as

$$S_K^C = \frac{V(E(u(\xi)|\xi_K))}{V(u(\xi))}, \quad S_C^T = \frac{E(V(u(\xi)|\xi_{K^c}))}{V(u(\xi))}, \tag{2.39}$$

where

$$E(u(\xi)|\xi_K) = \sum_{\substack{1\le i_k\le r_k \\ k\in K}} \beta_{(i_k)_{k\in K}}^K \bigotimes_{k\in K} v_{i_k}^k(\xi_k) \tag{2.40}$$

with $\beta_{(i_k)_{k\in K}}^K = \sum_{\substack{1\le i_l\le r_l \\ l\in K^c}} \alpha_i \prod_{l\in K^c} E(v_{i_l}^l(\xi_l))$.

# Chapter 3

# Sparse low rank tensor formats

*In chapter 2, we introduced several low rank tensor formats and their parametrization. We also briefly outlined approximation of tensors using approaches based on singular value decomposition and alternating least squares. In this chapter, we will define the corresponding sparse low rank tensor formats in section 1. Approximating a multivariate function in a sparse low rank tensor format using sampling based approach can be obtained using least squares with sparse regularization techniques which will be introduced in section 2. Alternating least square algorithm for performing approximation in sparse low rank tensor formats is presented in section 3. Algorithms for optimal selection of model parameters will be presented in section 4.*

## Contents

# 1 Sparse low rank tensor formats

Let us consider a low rank tensor subset $\mathcal{M}$ defined by

$$\mathcal{M} = \{v = F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m) : \mathbf{v}^i \in \mathcal{P}_{\mathcal{M}}, 1 \leq i \leq m\}, \tag{3.1}$$

where $F_{\mathcal{M}}$ is a multilinear map which constitutes the parametrization of the subset $\mathcal{M}$ and $\mathbf{v}^i$, $1 \leq i \leq m$, are the parameters (see chapter 2 section 3). We introduce the vectorization map $\mathcal{V} : \mathbf{v}^i \in \mathcal{P}^i_{\mathcal{M}} \mapsto \mathcal{V}(\mathbf{v}^i) \in \mathbb{R}^{\dim(\mathcal{P}_{\mathcal{M}})}$ such that $\mathcal{V}(\mathbf{v}^i)$ is a vector obtained by stacking the entries of $\mathbf{v}^i$ in a vector.

**Example 3.1.** *For the parametrization of $\mathcal{R}_r$ such that $v = F_{\mathcal{R}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$ with $\mathbf{v}^k \in \mathcal{P}^k_{\mathcal{R}_r} = \mathbb{R}^{n_k \times r}, 1 \leq k \leq d$, we have $\mathcal{V}(\mathbf{v}^k) \in \mathbb{R}^{n_k r}$ where the $i+1$-th column of $\mathbf{v}^k$ is stacked below $i$-th column.*

The corresponding sparse low rank tensor subset $\mathcal{M}^{\boldsymbol{m}\text{-sparse}}$ is defined as

$$\mathcal{M}^{\boldsymbol{m}\text{-sparse}} = \{v = F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m); \mathbf{v}^i \in \mathcal{P}^i_{\mathcal{M}}, \ \|\mathcal{V}(\mathbf{v}^i)\|_0 \leq m_i; 1 \leq i \leq m\}, \tag{3.2}$$

where $\|\cdot\|_0$ is the "$\ell_0$-norm" counting the number of non zero coefficients. Note that $\mathcal{M}^{\boldsymbol{m}\text{-sparse}}$ may have a very small effective dimension $\sum_{i=1}^m m_i \ll \sum_{i=1}^m n_i$.

The interest of defining sparse low rank tensor subsets is that they incorporate sparsity within the low rank structure thus enabling to exploit both low rank and sparsity structures of multivariate functions. However, solving a best approximation problem in $\mathcal{M}^{\boldsymbol{m}\text{-sparse}}$ leads to a combinatorial problem. Therefore, we replace the ideal sparse tensor subset $\mathcal{M}^{\boldsymbol{m}\text{-sparse}}$ by

$$\mathcal{M}^{\gamma} = \{v = F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m); \ \mathbf{v}^i \in \mathcal{P}^i_{\mathcal{M}}, \|\mathcal{V}(\mathbf{v}^i)\|_1 \leq \gamma_i, 1 \leq i \leq m\},$$

where we introduce a convex regularization of the constraints using $\ell_1$-norm [2, 13]. In practice, optimal approximations in subset $\mathcal{M}^{\gamma}$ can be computed using an alternating least squares algorithm that exploits the specific low dimensional parametrization of the subset $\mathcal{M}^{\gamma}$ and that involves the solution of successive least-squares problems with sparse $\ell_1$-regularization. Note that the dimension of $\mathcal{M}^{\gamma}$ is the same as the dimension of $\mathcal{M}$. However, approximations in $\mathcal{M}^{\gamma}$ obtained by using minimization may present sparsity.

# 2 Least squares methods

The objective of this section is to introduce notions associated with discrete least-squares with sparsity constraint. We will first consider approximations in classical function spaces in which we will introduce sparse regularization and the selection of regularization coefficient. We will then use these concepts to perform

approximations in sparse low rank tensor subsets in section 3.

We here consider the case of a real-valued model output $u : \Xi \to \mathbb{R}$. We denote by $\{y^q\}_{q=1}^Q \subset \Xi$ a set of $Q$ samples of $\xi$, and by $\{u(y^q)\}_{q=1}^Q \subset \mathbb{R}$ the corresponding function evaluations. We suppose that an approximation space $\mathcal{S}_P = \text{span}\{\phi_i\}_{i=1}^P$ is given. Classical least-squares method for the construction of an approximation $u_P \in \mathcal{S}_P$ then consists in solving the following problem:

$$\|u - u_P\|_Q^2 = \min_{v \in \mathcal{S}_P} \|u - v\|_Q^2 \quad \text{with} \quad \|u\|_Q^2 = \frac{1}{Q} \sum_{q=1}^{Q} u(y^q)^2. \tag{3.3}$$

Note that $\|\cdot\|_Q$ only defines a semi-norm on $L^2_{P_\xi}(\Xi)$ but it may define a norm on the finite dimensional subspace $\mathcal{S}_P$ if we have a sufficient number $Q$ of model evaluations. A necessary condition is $Q \geq P$. However, this condition may be unreachable in practice for high dimensional stochastic problems and usual a priori (non adapted) construction of approximation spaces $\mathcal{S}_P$. Moreover, classical least-squares method may yield bad results because of ill-conditioning (solution very sensitive to samples).

**Remark 3.1 :**
*The analysis in [19] indicates that for stochastic models containing uniformly distributed random variables and when using polynomial approximation, a sufficient condition to obtain a stable approximation is to choose a number of samples scaling quadratically with the dimension of polynomial space (i.e. $Q \propto P^2$).*

A way to circumvent these issues is to introduce a regularized least-squares functional:

$$\mathcal{J}^\lambda(v) = \|u - v\|_Q^2 + \lambda \mathcal{L}(v), \tag{3.4}$$

where $\mathcal{L}$ is a regularization functional and where $\lambda$ refers to some regularization parameter. The regularized least-squares problem then consists in solving

$$\mathcal{J}^\lambda(u_P^\lambda) = \min_{v \in \mathcal{S}_P} \mathcal{J}^\lambda(v). \tag{3.5}$$

Denoting by $\mathbf{v} = (v_1, \ldots, v_P)^T \in \mathbb{R}^P$ the coefficients of an element $v = \sum_{i=1}^P v_i \phi_i \in \mathcal{S}_P$, we can write

$$\|u - v\|_Q^2 = \|\mathbf{z} - \mathbf{\Phi}\mathbf{v}\|_2^2, \tag{3.6}$$

with $\mathbf{z} = (u(y^1), \ldots, u(y^Q))^T \in \mathbb{R}^Q$ the vector of random evaluations of $u(\xi)$ and $\mathbf{\Phi} \in \mathbb{R}^{Q \times P}$ the matrix with components $(\mathbf{\Phi})_{q,i} = \phi_i(y^q)$. We can then introduce a function $L : \mathbb{R}^P \to \mathbb{R}$ such that $\mathcal{L}(\sum_i v_i \phi_i) = L(\mathbf{v})$, and a function $J^\lambda : \mathbb{R}^P \to \mathbb{R}$

such that $\mathcal{J}^\lambda(\sum_i v_i\phi_i) = J^\lambda(\mathbf{v}) = \|\mathbf{z} - \mathbf{\Phi v}\|_2^2 + \lambda L(\mathbf{v})$. An algebraic version of least-squares problem (3.5) can then be written as follows:

$$\min_{\mathbf{v}\in\mathbb{R}^P} \|\mathbf{z} - \mathbf{\Phi v}\|_2^2 + \lambda L(\mathbf{v}). \tag{3.7}$$

Regularization introduces additional information such as smoothness, sparsity, etc. Under some assumptions on the regularization functional $\mathcal{L}$, problem (3.5) may have a unique solution. However, the choice of regularization strongly influences the quality of the obtained approximation. Another significant component of solving (3.7) is the choice of regularization parameter $\lambda$. In this thesis, we use cross validation for the selection of an optimal value of $\lambda$ (see section 4.1).

## 2.1 Sparse regularization

Over the last decade, sparse approximation methods have been extensively studied in different scientific disciplines. A sparse function is one that can be represented using few non zero terms when expanded on a suitable basis. If a function is known to be sparse on a particular basis, e.g. polynomial chaos (or tensor basis), sparse regularization methods can be used for quasi optimal recovery with only few sample evaluations. In general, a successful reconstruction of sparse solution vector depends on sufficient sparsity of the coefficient vector and additional properties (such as incoherence) depending on the samples and of the chosen basis (see [14, 27] or [29] in the context of uncertainty quantification). This strategy has been found to be effective for non-adapted sparse approximation of the solution of some PDEs [9, 29].

More precisely, an approximation $\sum_{i=1}^P u_i\phi_i(\xi)$ of a function $u(\xi)$ is considered as sparse on a particular basis $\{\phi_i(\xi)\}_{i=1}^P$ if it admits a good approximation with only a few non zero coefficients. Under certain conditions, a sparse approximation can be computed accurately using only $Q \ll P$ random samples of $u(\boldsymbol{\xi})$ via sparse regularization.
Given the random samples $\mathbf{z} \in \mathbb{R}^Q$ of the function $u(\xi)$, a best $m$-sparse (or $m$-term) approximation of $u$ can be ideally obtained by solving the constrained optimization problem

$$\min_{\mathbf{v}\in\mathbb{R}^P} \|\mathbf{z} - \mathbf{\Phi v}\|_2^2 \quad \text{subject to} \quad \|\mathbf{v}\|_0 \leq m, \tag{3.8}$$

where $\|\mathbf{v}\|_0 = \#\{i \in \{1,\dots,P\} \ : \ v_i \neq 0\}$ is the so called "$\ell_0$-norm" of $\mathbf{v}$ which gives the number of non zero components of $\mathbf{v}$. Problem (3.8) is a combinatorial optimization problem which is NP hard to solve. Under certain assumptions, problem (3.8) can be reasonably well approximated by the following constrained optimization problem which introduces a convex relaxation of the "$\ell_0$-norm":

$$\min_{\mathbf{v}\in\mathbb{R}^P} \|\mathbf{z} - \mathbf{\Phi v}\|_2^2 \quad \text{subject to} \quad \|\mathbf{v}\|_1 \leq \delta, \tag{3.9}$$

where $\|\mathbf{v}\|_1 = \sum_{i=1}^{P} |v_i|$ is the $\ell_1$-norm of $\mathbf{v}$. Since $\ell_1$-norm is convex, we can equivalently consider the following convex optimization problem, known as Lasso [74] or basis pursuit [16]:

$$\min_{\mathbf{v}\in\mathbb{R}^P} \|\mathbf{z} - \mathbf{\Phi}\mathbf{v}\|_2^2 + \lambda\|\mathbf{v}\|_1, \tag{3.10}$$

where $\lambda > 0$ corresponds to a Lagrange multiplier whose value is related to $\delta$. Problem (3.10) appears as a regularized least-squares problem. The $\ell_1$-norm is a sparsity inducing regularization function in the sense that the solution $\mathbf{v}$ of (3.10) may contain components which are exactly zero. Several optimization algorithms have been proposed for solving (3.10) ( see [2] for a comprehensive review).

# 3 Approximation in sparse low rank tensor formats

## 3.1 ALS for sparse low rank tensor formats

We now have all the ingredients to formulate approximation problem in sparse low rank tensor format using least squares. We first consider approximation of a real valued multivariate function $u : \Xi \to \mathbb{R}$ in a low rank subset $\mathcal{M}$ (such as canonical or tree based formats). The discrete least-squares minimization problem then takes the form

$$\min_{\mathbf{v}^1\in\mathcal{P}^1_{\mathcal{M}},\ldots,\mathbf{v}^m\in\mathcal{P}^m_{\mathcal{M}}} \|u - F_{\mathcal{M}}(\mathbf{v}^1,\ldots,\mathbf{v}^m)\|_Q^2, \tag{3.11}$$

where the function to minimize is quadratic with respect to each argument $\mathbf{v}^i$, $1 \leq i \leq m$. We can thus use alternating least-squares algorithm (see chapter 2 section 4.2.).
Minimization in $\mathcal{M}$ exploits the low rank structure of $u(\xi)$. However, when the number of available samples is not sufficient to obtain stable estimation of $\mathbf{v}^i$, $1 \leq i \leq m$, we can perform optimization in the corresponding sparse low rank tensor subset $\mathcal{M}^\gamma$. Thus within ALS, we formulate least squares problem with sparse regularization given by

$$\min_{\mathbf{v}^1\in\mathcal{P}^1_{\mathcal{M}},\ldots,\mathbf{v}^m\in\mathcal{P}^m_{\mathcal{M}}} \|u - F_{\mathcal{M}}(\mathbf{v}^1,\ldots,\mathbf{v}^m)\|_Q^2 + \sum_{i=1}^{m} \lambda_i\|\mathcal{V}(\mathbf{v}^i)\|_1, \tag{3.12}$$

where the regularization parameter can be selected using cross validation (see section 4). Algorithm 4 outlines the ALS algorithm for approximation in sparse low rank tensor format.

**Algorithm 4** Alternating Least Square with sparsity inducing regularization

**Input:** $\{y^q\}_{q=1}^Q$ samples of $\xi$ and model evaluations $\{u(y^q)\}_{q=1}^Q$.
**Output:** $v \in \mathcal{M}$.

1: Initialize $(\mathbf{v}^1, \ldots, \mathbf{v}^m)$ randomly;
2: $v = F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m)$;
3: **while** $v$ has not converge **do**
4: **for** $i = 1, \ldots, m$ **do**
5: $\mathbf{v}^i = \underset{\mathbf{w} \in \mathcal{P}^i_{\mathcal{M}}}{\operatorname{argmin}} \|u - F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{w}, \ldots, \mathbf{v}^m)\|_Q^2 + \lambda_i \|\mathcal{V}(\mathbf{w})\|_1$ with selection of $\lambda_i$ through cross validation .
6: $v = F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m)$;
7: **end for**
8: **end while**
9: **return** $v$;

Approximation obtained by using algorithm 4 for a given number of samples of the function $u$ depends on the specific low rank tensor format considered. For a given sample set, we can obtain approximations of the same function in different sparse low rank tensor formats with varying level of accuracy.
In many practical cases, we can choose sparse canonical tensor format $\mathcal{R}_r$. This format has few real parameters to estimate and may give good approximation with very few sample evaluations. Convergence of algorithm 4 in canonical format is discussed in Chapter 4. Approximation in sparse Tucker tensor format $\mathcal{T}_r^\gamma$ involves minimization over the core tensor which may not be feasible for high dimensional functions $u$ or functions with high multilinear rank. Sparse Hierarchical format $\mathcal{H}_r^T$ is a good candidate but involves the selection of a binary tree $T$ for which we have combinatorial possibilities. In this case, we need a priori information about the function $u$ to determine optimal regrouping of variables in tree nodes, or an adaptive strategy to construct the tree. A simple tree based representation is the Tensor Train format $\mathcal{TT}_r^\gamma$, which results in good approximation in many practical cases. In chapter 5, we study algorithm 4 in $\mathcal{TT}^\gamma$.

# 4 Cross validation for selection of model parameters

Approximation of functions using regularized least squares requires a selection of model parameters. Cross validation is a standard statistical technique for the selection of these parameters using the available information.

Cross validation consists of randomly partitioning the available samples $Q$ into two sub-samples, the *training set* and the *test set*. We choose several values of model parameters and solve for their corresponding models using training set and assess the accuracy of the solution by predicting its error on the test set. A

refinement of this method is to use $k$-fold cross validation in which the samples are randomly assigned to one of the $k$-sets of approximately equal size in a partition. The training set consists of all but one of the sets of the partition which is considered as the test set. The model error is estimated for each of the $k$ sets and averaged over the $k$ sets. We then select the optimal model corresponding to the parameter for which model error is minimum. A leave one out error is a special case of cross validation in which $k = Q$.

In this thesis, we use cross validation model selection for the selection of regularization coefficient $\lambda$ in solving regularised least squares problem and for the selection of optimal tensor rank $r$ in sparse low rank tensor format while using algorithm 4.

## 4.1 Selection of regularization coefficient $\lambda_i$

An important component of algorithm 4 is the selection of regularization coefficient $\lambda_i$ in 5. A small value of $\lambda_i$ may lead to over-fitting whereas a larger value may lead to stronger sparsity constraint thus deteriorating the solution. Therefore optimal value of $\lambda_i$ can be estimated using cross validation. However, this method has a disadvantage when several values of $\lambda_i$ are being considered as it involves training the model for each value of $\lambda_i$. Moreover in guessing these values, one can miss optimal $\lambda_i$ thus leading to a suboptimal solution.

In this thesis, we use the Lasso modified least angle regression algorithm (see LARS presented in [30]). The advantage of this algorithm is that it provides a set of $N_r$ solutions, namely the regularization path, with increasing $\ell_1$-norm. Thus it eliminates the need to explicitly choose the values of $\lambda_i$. Let $\mathbf{v}^j$, with $j = 1, \ldots, N_r$, denote this set of solutions, $A_j \subset \{1, \ldots, P\}$ be the index set corresponding to non zero coefficients of $\mathbf{v}^j$, $\mathbf{v}^j_{A_j} \in \mathbb{R}^{\#A_j}$ the vector of the coefficients $A_j$ of $\mathbf{v}^j$, and $\mathbf{\Phi}_{A_j} \in \mathbb{R}^{Q \times \#A_j}$ the submatrix of $\mathbf{\Phi}$ obtained by extracting the columns of $\mathbf{\Phi}$ corresponding to indices $A_j$. The optimal solution $\mathbf{v}$ is then selected using the fast leave-one-out cross validation error estimate [15] which relies on the use of the Sherman-Morrison-Woodbury formula (see [9] for its implementation within Lasso modified LARS algorithm). Algorithm 5 briefly outlines the cross validation procedure for the selection of the optimal solution. In this work, we have used Lasso modified LARS implementation of SPAMS software [50] for $\ell_1$-regularization.

## 4.2 Selection of optimal tensor rank

Let us note that algorithm 4 works in a given low rank format with fixed rank. A simple approach can be used to select the rank. In sampling based approach, the optimal rank may depend on the number of samples $Q$ and the sample set. Hence we need a model selection technique to select the optimal rank. In this work, we

**Algorithm 5** Algorithm to determine optimal LARS solution using leave-one-out cross validation.

**Input:** sample vector $\mathbf{z} \in \mathbb{R}^Q$ and matrix $\mathbf{\Phi} \in \mathbb{R}^{Q \times P}$
**Output:** vector of coefficients $\mathbf{v} \in \mathbb{R}^P$

1: Run the Lasso modified LARS procedure to obtain $N_r$ solutions $\mathbf{v}^1, \ldots, \mathbf{v}^{N_r}$ of the regularization path, with corresponding sets of non zeros coefficients $A_1, \ldots, A_{N_r}$.
2: **for** $j = 1, \ldots, N_r$ **do**
3: Recompute the non zero coefficients $\mathbf{v}^j_{A_j}$ of $\mathbf{v}^j$ using ordinary least-squares: $\mathbf{v}^j_{A_j} = \arg\min_{\mathbf{v} \in \mathbb{R}^{\#A_j}} \|\mathbf{z} - \mathbf{\Phi}_{A_j}\mathbf{v}\|_2^2$
4: Compute $h_q = (\mathbf{\Phi}_{A_j}(\mathbf{\Phi}^T_{A_j}\mathbf{\Phi}_{A_j})^{-1}\mathbf{\Phi}^T_{A_j})_{qq}$.
5: Compute relative leave-one-out error $\epsilon_j = \frac{1}{Q}\sum_{q=1}^{Q}\left(\frac{(\mathbf{z})_q - (\mathbf{\Phi}_{A_j}\mathbf{v}^j_{A_j})_q}{(1-h_q)\hat{\sigma}(\mathbf{z})}\right)^2$, where $\hat{\sigma}(\mathbf{z})$ is the empirical standard deviation of $\mathbf{z}$.
6: **end for**
7: Return the optimal solution $\mathbf{v}$ such that $\mathbf{v}_{A_{j^*}} = \mathbf{v}^{j^*}_{A_{j^*}}$ with $j^* = \arg\min_j \epsilon_j$.

use a $k$-fold cross validation method. The overall procedure is as follows:

- Choose several values of tensor rank $r_j$, $1 \leq j \leq J$.
- Split sample set $S = \{1, \ldots, Q\}$ into $k$ disjoint subsamples $\{G_i\}_{i=1}^k$, $G_i \subset S$, of approximately the same size, and let $S_i = S \setminus G_i$.
- For each subsample $S_i$ and tensor rank $r_j$, run Algorithm 4 to obtain (approximated) model $v^i_{r_j}$. Compute the corresponding mean squared errors $\{\varepsilon^{S_i}_{r_1}, \ldots, \varepsilon^{S_i}_{r_j}\}$ from the test set $G_i$.
- Compute the $k$-fold cross validation error $\bar{\varepsilon}_{r_j} = \frac{1}{k}\sum_{i=1}^k \varepsilon^{S_i}_{r_j}, j = 1, \ldots, J$.
- Select optimal rank $r_{op}$ such that $\epsilon_{r_{op}} = \min_{1 \leq j \leq J} \epsilon_{r_j}$.
- Run Algorithm 4 using a rank $r_{op}$ with the whole data set $S$ for computing $v_{r_{op}}$.

# 5 Approximation error and sparsity ratio

In the following chapters, for the purpose of estimating the approximation error in sparse low rank formats, we introduce the relative error $\varepsilon(u_r, u)$ between the function $u$ and its low rank approximation $u_r$, estimated with Monte Carlo integration with $Q'$ samples:

$$\varepsilon(u_r, u) = \frac{\|u_r - u\|_{Q'}}{\|u\|_{Q'}}. \tag{3.13}$$

We also define the *total sparsity ratio* $\varrho$ of a low rank approximation $v = F_{\mathcal{M}}(\mathbf{v}^1, \ldots, \mathbf{v}^m)$ as:

$$\varrho = \frac{\sum_{i=1}^{m} \|\mathcal{V}(\mathbf{v}^i)\|_0}{\sum_{i=1}^{m} \dim(\mathcal{P}^i_{\mathcal{M}})}, \tag{3.14}$$

where $\|\mathcal{V}(\mathbf{v}^i)\|_0$ gives the number of *non zero* real coefficients in the parameter $\mathbf{v}^i$ and $\dim(\mathcal{P}^i_{\mathcal{M}})$ gives the total number of real coefficients in $\mathbf{v}^i$. In short, $\varrho$ is the ratio of total number of non zero parameters to the total number of parameters in a low rank tensor representation. We also define the *partial sparsity ratio* $\varrho_i$ as

$$\varrho^{(i)} = \frac{\|\mathcal{V}(\mathbf{v}^i)\|_0}{\dim(\mathcal{P}^i_{\mathcal{M}})} \tag{3.15}$$

which indicates the sparsity ratio of parameter $\mathbf{v}^i$. These values will be used for analysing approximation and sparsity in different low rank tensor formats in the illustrative examples.

# Chapter 4

# Approximation in Sparse Canonical Tensor Format

*In this chapter, we describe the implementation of approximation in sparse canonical tensor format. In section 1, we introduce sparse canonical tensor format and their parametrization. We then present Alternating Least-Squares (ALS) algorithm for construction of sparse rank one approximation in section 2 followed by an updated greedy algorithm for approximation in sparse canonical rank-m format in section 4. We illustrate these algorithms on numerical examples in section 5. Finally in section 6, we extend the applicability of these ideas for vector valued functions.*

## Contents

The aim in this chapter is to find an approximation of a function $u(\xi)$ in the finite dimensional tensor space $\mathcal{S}_{\boldsymbol{n}} = \mathcal{S}^1_{n_1} \otimes \ldots \otimes \mathcal{S}^d_{n_d}$ in sparse low rank canonical tensor format. In the proposed method, approximations are computed in small low-rank tensor subsets using least-squares with sparsity inducing regularization. Here, we first present the case where successive corrections are computed in the elementary set of rank-one tensors $\mathcal{R}_1$, thus resulting in the construction of a rank one canonical approximation of the solution. We then present an updated greedy algorithm for performing approximation in tensor subset $\mathcal{R}_m$.

# 1 Sparse canonical tensor subsets

Let $\mathcal{R}_1$ denote the set of (elementary) rank-one tensors in $\mathcal{S}_{\boldsymbol{n}} = \mathcal{S}^1_{n_1} \otimes \ldots \otimes \mathcal{S}^d_{n_d}$, defined by

$$\mathcal{R}_1 = \left\{ v(y) = \left(\otimes_{k=1}^d v^k\right)(y) = \prod_{k=1}^d v^k(y_k) \ ; \ v^k \in \mathcal{S}^k_{n_k} \right\},$$

or equivalently by

$$\mathcal{R}_1 = \left\{ v(y) = \langle \boldsymbol{\phi}(y), \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle ; \mathbf{v}^k \in \mathbb{R}^{n_k} \right\},$$

where $\boldsymbol{\phi}(y) = \boldsymbol{\phi}^1(y_1) \otimes \ldots \otimes \boldsymbol{\phi}^d(y_d)$, with $\boldsymbol{\phi}^k = (\phi^k_1, \ldots, \phi^k_{n_k})^T$ the vector of basis functions of $\mathcal{S}^k_{n_k}$, and where $\mathbf{v}^k = (v^k_1, \ldots, v^k_{n_k})^T$ is the set of coefficients of $v^k$ in the basis of $\mathcal{S}^k_{n_k}$, that means $v^k(y_k) = \sum_{i=1}^{n_k} v^k_i \phi^k_i(y_k)$.
Approximation in $\mathcal{R}_1$ using classical least-squares methods possibly enables to recover a good approximation of the solution using a reduced number of samples. However it may not be sufficient in the case where the approximation spaces $\mathcal{S}^k_{n_k}$ have high dimensions $n_k$, thus resulting in a manifold of rank-one elements $\mathcal{R}_1$ with high dimension $\sum_{k=1}^d n_k$. This difficulty may be circumvented by introducing approximations in the corresponding $\boldsymbol{m}$-sparse rank-one subset defined as

$$\mathcal{R}_1^{\boldsymbol{m}\text{-sparse}} = \left\{ v(y) = \langle \boldsymbol{\phi}(y), \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle ; \mathbf{v}^k \in \mathbb{R}^{n_k}, \|\mathbf{v}^k\|_0 \leq m_k \right\}$$

with effective dimension $\sum_{k=1}^d m_k \ll \sum_{k=1}^d n_k$. As mentioned in section 2.1, performing least-squares approximation in this set may not be computationally tractable. We thus introduce a convex relaxation of the $\ell_0$-"norm" to define the subset $\mathcal{R}_1^\gamma$ of $\mathcal{R}_1$ defined as

$$\mathcal{R}_1^\gamma = \left\{ v(y) = \langle \boldsymbol{\phi}(y), \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle ; \mathbf{v}^k \in \mathbb{R}^{n_k}, \|\mathbf{v}^k\|_1 \leq \gamma_k \right\},$$

where the set of parameters $(\mathbf{v}^1, \ldots, \mathbf{v}^d)$ is now searched in a convex subset of $\mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_d}$.

Finally, we introduce the set of canonical rank-$m$ tensors $\mathcal{R}_m = \{w = \sum_{i=1}^m v_i \, ; v_i \in \mathcal{R}_1\}$ and the corresponding subset

$$\mathcal{R}_m^{\gamma^1,\ldots,\gamma^m} = \left\{ w = \sum_{i=1}^m v_i \, ; v_i \in \mathcal{R}_1^{\gamma^i} \right\}.$$

In the following, we propose algorithms for the construction of approximations in tensor subsets $\mathcal{R}_1^\gamma$ and $\mathcal{R}_m^{\gamma^1,\ldots,\gamma^m}$.

# 2 Construction of sparse rank-one tensor approximation

The subset $\mathcal{R}_1^\gamma$ can be parametrized with the set of parameters $(\mathbf{v}^1, \ldots, \mathbf{v}^d) \in \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_d}$ such that $\|\mathbf{v}^k\|_1 \leq \gamma_k$ $(k = 1, \ldots, d)$, this set of parameters corresponding to an element $v = \otimes_{k=1}^d v^k$ where $\mathbf{v}^k$ denotes the vector of coefficients of an element $v^k$. With appropriate choice of bases, a sparse rank-one function $v$ could be well approximated using vectors $\mathbf{v}^k$ with only a few non zero coefficients.
We compute a rank-one approximation $v = \otimes_{k=1}^d v^k \in \mathcal{R}_1^\gamma$ of $u$ by solving the least-squares problem

$$\min_{v \in \mathcal{R}_1^\gamma} \|u - v\|_Q^2 = \min_{\substack{\mathbf{v}^1 \in \mathbb{R}^{n_1}, \ldots, \mathbf{v}^d \in \mathbb{R}^{n_d} \\ \|\mathbf{v}^1\|_1 \leq \gamma_1, \ldots, \|\mathbf{v}^d\|_1 \leq \gamma_d}} \|u - \langle \boldsymbol{\phi}, \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle\|_Q^2. \tag{4.1}$$

Problem (4.1) can be equivalently written

$$\min_{\mathbf{v}^1 \in \mathbb{R}^{n_1}, \ldots, \mathbf{v}^d \in \mathbb{R}^{n_d}} \|u - \langle \boldsymbol{\phi}, \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle\|_Q^2 + \sum_{k=1}^d \lambda_k \|\mathbf{v}^k\|_1, \tag{4.2}$$

where the values of the regularization parameters $\lambda_k > 0$ (interpreted as Lagrange multipliers) are related to $\gamma_k$. In practice, minimization problem (4.2) is solved using an alternating minimization algorithm which consists in successively minimizing over $\mathbf{v}^j$ for fixed values of $\{\mathbf{v}^k\}_{k \neq j}$. Denoting by $\mathbf{z} \in \mathbb{R}^Q$ the vector of samples of function $u(\xi)$ and by $\boldsymbol{\Phi}^j \in \mathbb{R}^{Q \times n_j}$ the matrix whose components are $(\boldsymbol{\Phi}^j)_{qi} = \phi_i^j(y_j^q) \prod_{k \neq j} v^k(y_k^q)$, the minimization problem on $v^j$ can be written

$$\min_{\mathbf{v}^j \in \mathbb{R}^{n_j}} \|\mathbf{z} - \boldsymbol{\Phi}^j \mathbf{v}^j\|_2^2 + \lambda_j \|\mathbf{v}^j\|_1, \tag{4.3}$$

which has a classical form of a least-squares problem with a sparsity inducing $\ell_1$-regularization. Problem (4.3) is solved using the Lasso modified LARS algorithm where the optimal solution is selected using the leave-one-out cross validation procedure presented in Algorithm 5 in chapter 3. Algorithm 6 outlines the construction of a sparse rank one approximation.

**Algorithm 6** Algorithm to compute sparse rank one approximation of a function $u$.

**Input:** vector of evaluations $\mathbf{z} = (u(y^1), \ldots, u(y^Q))^T \in \mathbb{R}^Q$.
**Output:** rank-one approximation $v(y) = \langle \boldsymbol{\phi}(y), \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle$.

1: Set $l = 0$ and initialize the vectors $\{\mathbf{v}^k\}_{k=1}^d$ of $v_l = \langle \boldsymbol{\phi}(y), \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle$.
2: $l \leftarrow l + 1$.
3: **for** $j = 1, \ldots, d$ **do**
4: Evaluate matrix $\boldsymbol{\Phi}^j$.
5: Solve problem (4.3) using Algorithm 5 for input $\mathbf{z} \in \mathbb{R}^Q$ and $\boldsymbol{\Phi}^j \in \mathbb{R}^{Q \times n_j}$ to obtain $\mathbf{v}^j$.
6: **end for**
7: Set $v_l = \langle \boldsymbol{\phi}(y), \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle$
8: **if** $\|v_l - v_{l-1}\| > \epsilon$ and $l \leq l_{max}$ **then**
9: Go to Step 2.
10: **end if**
11: Return the rank one approximation $v = v_l$.

**Remark 4.1 (Other types of regularization) :**
*Different rank-one approximations can be defined by changing the type of regularization and constructed by replacing step 5 of Algorithm 6. First, one can consider ordinary least squares by replacing step 5 by the solution of*

$$\min_{\mathbf{v}^j \in \mathbb{R}^{n_j}} \|\mathbf{z} - \boldsymbol{\Phi}^j \mathbf{v}^j\|_2^2. \tag{4.4}$$

*Also, one can consider a regularization using $\ell_2$-norm (ridge regression) by replacing step 5 by the solution of*

$$\min_{\mathbf{v}^j \in \mathbb{R}^{n_j}} \|\mathbf{z} - \boldsymbol{\Phi}^j \mathbf{v}^j\|_2^2 + \lambda_j \|\mathbf{v}^j\|_2^2, \tag{4.5}$$

*with a selection of optimal parameter $\lambda_j$ using standard cross-validation (typically k-fold cross-validation). The approximations obtained with these different variants will be compared in the numerical examples of section 5.*

# 3 Direct construction of sparse rank-$m$ tensor approxmation

Algorithm 7 extends algorithm 6 for direct approximation in $\mathcal{R}_m$.Best approximation problem in $\mathcal{R}_m$ for $m \geq 2$ and $d > 2$ is an ill-posed problem since $\mathcal{R}_m$ is not closed (see e.g. [39] lemma 9.11 pg 255). However, using sparsity-inducing regularization (and also other types of regularizations) makes the best approximation problem well-posed. Indeed, it can be proven that the subset $\mathcal{R}_m^{\gamma^1,\ldots,\gamma^m}$ of canonical tensors with bounded factors is a closed subset. For the construction of a rank-$m$

approximation, we will however introduce another algorithm (see section 4) which is based on progressive construction of successive rank-one corrections.

**Algorithm 7** Algorithm to compute direct sparse rank-$m$ approximation of a function $u$.

**Input:** vector of evaluations $\mathbf{z} = (u(y^1), \ldots, u(y^Q))^T \in \mathbb{R}^Q$, rank $m$.
**Output:** A rank $m$ approximation $u_m(y) = \langle \boldsymbol{\phi}(y), \sum_{i=1}^m \mathbf{v}_i^1 \otimes \ldots \otimes \mathbf{v}_i^d \rangle$

1: Initialize the vectors $\{\mathbf{v}_i^k\}_{k=1}^d, 1 \leq i \leq m$ and set $u_m^l = \langle \boldsymbol{\phi}(y), \sum_{i=1}^m \mathbf{v}_i^1 \otimes \ldots \otimes \mathbf{v}_i^d \rangle$ with $l = 0$
2: Set $l \leftarrow l + 1$
3: **for** $j = 1, \ldots, d$ **do**
4: Evaluate the matrix $\boldsymbol{\Phi}^j = [\boldsymbol{\Phi}_1^j \cdots \boldsymbol{\Phi}_m^j]$, $\boldsymbol{\Phi}_i^j \in \mathbb{R}^{Q \times n_j}$, $(\boldsymbol{\Phi}_i^j)_{qj'} = \phi_{j'}^j(y_j^q) \prod_{k \neq j} v_i^k(y_k^q), 1 \leq j' \leq n_j$.
5: Solve $\min_{\mathbf{v}^j \in \mathbb{R}^{n_j \times m}} \|\mathbf{z} - \boldsymbol{\Phi}^j \mathcal{V}(\mathbf{v}^j)\|_2^2 + \lambda_j \|\mathcal{V}(\mathbf{v}^j)\|_1$ with selection of $\lambda_j$ using cross validation.
6: **end for**
7: Set $u_m^l = \langle \boldsymbol{\phi}(y), \sum_{i=1}^m \mathbf{v}_i^1 \otimes \ldots \otimes \mathbf{v}_i^d \rangle$
8: **if** $\|u_m^l - u_m^{l-1}\| > \epsilon$ and $l \leq l_{max}$ **then**
9: Go to Step 2
10: **end if**
11: Return $u_m = u_m^l$.

# 4 Updated greedy construction of sparse rank-$m$ approximation

We now wish to construct a sparse rank-$m$ approximation $u_m \in \mathcal{R}_m$ of $u$ of the form $u_m = \sum_{i=1}^m \alpha_i v_i$ by successive computations of sparse rank-one approximations $v_i = \otimes_{k=1}^d v_i^k, 1 \leq i \leq m$.

**Remark 4.2 :**
*A pure greedy construction may yield a suboptimal rank-m approximation but it has several advantages: successive minimization problems in $\mathcal{R}_1$ are well-posed (without any regularization), it requires the estimation of a small number of parameters at each iteration.*

We start by setting $u_0 = 0$. Then, knowing an approximation $u_{m-1}$ of $u$, we proceed as follows.

## 4.1 Sparse rank-1 correction step

We first compute a correction $v \in \mathcal{R}_1$ of $u_{m-1}$ by solving

$$\min_{v \in \mathcal{R}_1^\gamma} \|u - u_{m-1} - v\|_Q^2, \tag{4.6}$$

which can be reformulated as

$$\min_{v \in \mathcal{R}_1} \|u - u_{m-1} - \langle \boldsymbol{\phi}, \mathbf{v}^1 \otimes \ldots \otimes \mathbf{v}^d \rangle\|_Q^2 + \sum_{k=1}^d \lambda_k \|\mathbf{v}^k\|_1. \tag{4.7}$$

Problem (4.7) is solved using an alternating minimization algorithm, which consists in successive minimization problems of the form (4.3) where $\mathbf{z} \in \mathbb{R}^Q$ is the vector of samples of the residual $(u - u_{m-1})(\xi)$. Optimal parameters $\{\lambda_k\}_{k=1}^d$ are selected with the fast leave-one-out cross-validation.

## 4.2 Updating step

Once a rank-one correction $v_m$ has been computed, it is normalized and the approximation $u_m = \sum_{i=1}^m \alpha_i v_i$ is computed by solving a regularized least-squares problem:

$$\min_{\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_m)\in\mathbb{R}^m} \|u - \sum_{i=1}^m \alpha_i v_i\|_Q^2 + \lambda' \|\boldsymbol{\alpha}\|_1. \tag{4.8}$$

This updating step allows a selection of significant terms in the canonical decomposition, that means when some $\alpha_i$ are found to be negligible, it yields an approximation $u_m = \sum_{i=1}^m \alpha_i v_i$ with a lower effective rank representation. The selection of parameter $\lambda'$ is also done with a cross-validation technique. Note that even if the $\ell_1$ regularization does not yield a selection of a subset of terms, it is still useful in the situation when a few samples are available.

**Remark 4.3 :**
*Note that an improved updating strategy could be introduced as follows. At step $m$, denoting by $v_i = \otimes_{k=1}^d v_i^k$, $1 \le i \le m$, the computed corrections, we can define approximation spaces $\mathcal{V}_m^k = span\{v_i^k\}_{i=1}^m \subset \mathcal{S}_{n_k}^k$ (with dimension at most $m$), and look for an approximation of the form $u_m = \sum_{i_1=1}^m \ldots \sum_{i_d=1}^m \alpha_{i_1\ldots i_d} \otimes_{k=1}^d v_{i_k}^k \in \otimes_{k=1}^d \mathcal{V}_m^k$ (Tucker tensor format).*
*The update problem then consists in solving*

$$\min_{\boldsymbol{\alpha}=(\alpha_{i_1\ldots i_d})\in\mathbb{R}^{m\times\ldots\times m}} \|u - \sum_{i_1,\ldots,i_d} \alpha_{i_1\ldots i_d} \otimes_{k=1}^d v_{i_k}^k\|_Q^2 + \lambda' \|\boldsymbol{\alpha}\|_1, \tag{4.9}$$

*where $\|\boldsymbol{\alpha}\|_1 = \sum_{i_1,\ldots,i_d} |\alpha_{i_1\ldots i_d}|$. This updating strategy can yield significant improvements of convergence. However, it is clearly unpractical for high dimension $d$ since the dimension $m^d$ of the core grows exponentially with $d$. For high dimension, other types of representations should be introduced, such as hierarchical tensor representations (see chapter 5).*

Algorithm 8 details the updated greedy construction of sparse low rank approximations.

**Algorithm 8** Updated greedy algorithm for sparse low rank approximation of a function $u$.

**Input:** vector of evaluations $\mathbf{z} = (u(y^1), \ldots, u(y^Q))^T \in \mathbb{R}^Q$ and maximal rank $M$.
**Output:** Sequence of approximations $u_m = \sum_{i=1}^m \alpha_i v_i$, where $v_i \in \mathcal{R}_1$ and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$.

1: Set $u_0 = 0$
2: **for** $m = 1, \ldots, M$ **do**
3: Evaluate the vector $\mathbf{z}_{m-1} = (u_{m-1}(y^1), \ldots, u_{m-1}(y^Q))^T \in \mathbb{R}^Q$
4: Compute a sparse rank-one approximation $v_m = \otimes_{k=1}^d v_m^k$ using Algorithm 6 for input vector of evaluations $\mathbf{z} - \mathbf{z}_{m-1}$.
5: Evaluate matrix $\boldsymbol{W} \in \mathbb{R}^{Q \times m}$ with components $(\boldsymbol{W})_{qi} = v_i(y^q)$.
6: Compute $\boldsymbol{\alpha} \in \mathbb{R}^m$ with Algorithm 5 for input vector $\mathbf{z} \in \mathbb{R}^Q$ and matrix $\boldsymbol{W} \in \mathbb{R}^{Q \times m}$.
7: **end for**
8: Set $u_m = \sum_{i=1}^m \alpha_i v_i$

# 5 Application examples

In this section, we validate the proposed algorithm on several benchmark problems. The purpose of the first example on Friedman function in section 5.1 is to highlight the benefit of the greedy low rank approximation by giving some hints on the number of samples needed for a stable approximation. The three following examples then exhibit the robustness of the $\ell_1$-regularization within the low rank approximation:

- by correctly detecting sparsity when appropriate approximation space is introduced as in the checker board function case presented in section 5.2,
- or just by looking for the simplest representation with respect to the number of samples as in the examples of sections 5.3 and 5.4.

## 5.1 Analytical model: Friedman function

Let us consider a simple benchmark problem namely the Friedman function of dimension $d = 5$ also considered in [4]:

$$u(\xi) = 10 \sin(\pi \xi_1 \xi_2) + 20(\xi_3 - 0.5)^2 + 10\xi_4 + 5\xi_5,$$

where $\xi_i, i = 1, \ldots, 5$, are uniform random variables over [0,1]. In this section, we consider low rank tensor subsets without any sparsity constraint, that means we do

not perform $\ell_1$ regularization in the alternating minimization algorithm. The aim is to estimate numerically the number of function evaluations necessary in order to obtain a stable approximation in low rank tensor format.

The analysis on the number of function evaluations is here based upon results of [19] where it is proven that, for a stable approximation of a univariate function with optimal convergence rate using ordinary least squares on polynomial spaces, the number of random sample evaluations required scales quadratically with the dimension of the polynomial space, i.e. $Q \sim (p+1)^2$ where $p$ is the maximal polynomial degree.

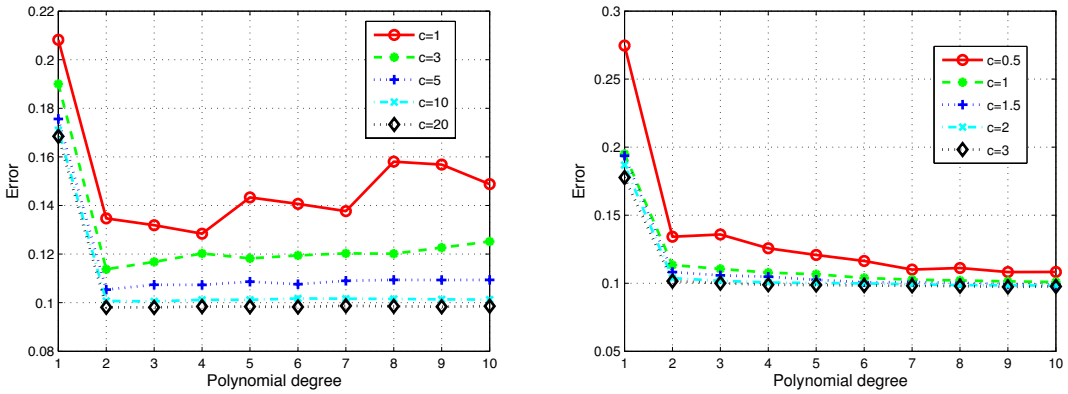
This result is supported with numerical tests on univariate functions and on multivariate functions that show that indeed choosing $Q$ scaling quadratically with the dimension of the polynomial space $P$ is robust while $Q$ scaling linearly with $P$ may lead to an unstable approximation, depending on $P$ and the dimension $d$. The following numerical tests aim at bringing out a similar type of rule for choosing the number of samples $Q$ for a low rank approximation of a multivariate function constructed in a greedy fashion according to Algorithm 8, but with ordinary least squares in step 5 of Algorithm 6, and given an isotropic tensor product polynomial approximation space with maximum degree $p$ in all dimensions. We first consider a rank one approximation of the function in $\mathbb{P}_p \otimes \cdots \otimes \mathbb{P}_p$ where $\mathbb{P}_p$ denotes the polynomial approximation space of maximal degree $p$ in each dimension from 1 to $d$. Given the features above and considering the algorithm for the construction of the rank one element of order $d$, we consider the following rule:

$$Q = cd(p+1)^\alpha$$

where $c$ is a positive constant and $\alpha = 1$ (linear rule) or 2 (quadratic rule). In the following analyses of the current section, we plot the mean $\varepsilon(u_m, u)$ (see (3.13)) over 51 sample set repetitions in order to reduce the dependence on the sample set of a given size. In figure 4.1, we compare the error of rank one approximation $\varepsilon(u_1, u)$ with respect to the polynomial degree $p$ using both linear rule (left) and quadratic rule (right) for different values of $c$ (ranging from 1 to 20 in the linear rule and 0.5 to 3 in quadratic rule). As could have been expected, we find that the linear rule yields a deterioration for small values of $c$ whereas the quadratic rule gives a stable approximation with polynomial degree. For higher rank approximations, the total number of samples needed will have a dependence on the approximation rank $m$. Thus we modify sample size estimates and consider the rule

$$Q = cdm(p+1)^\alpha$$

with $\alpha = 1$ (linear rule) or 2 (quadratic rule). In figure 4.2, we plot approximation error $\varepsilon(u_m, u)$ using linear rule (left) and quadratic rule (right) for $m = 2, 3, 4$ and different values of $c$. We find again that quadratic rule gives a stable approximation




(a) (b)

**Figure 4.1:** Friedman function: Evolution of rank one approximation error $\varepsilon(u_1, u)$ with respect to polynomial degree $p$ with (a) $Q = cd(p+1)$ and (b) $Q = cd(p+1)^2$ samples with several values of $c$.

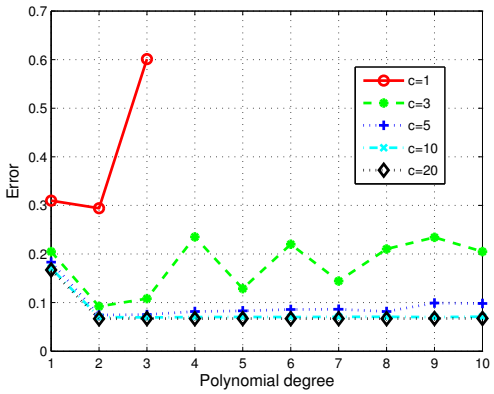
for $c \geq 1$.

In order to analyse the accuracy of the rank-$m$ approximation with respect to $m$, in figure 4.3 we plot the error $\varepsilon(u_m, u)$ with respect to the polynomial degree $p$ for different values of $m$ using $Q = dm(p+1)^2$, that is $c = 1$. As the number of samples $Q$ increases with rank $m$ using this rule, more information on the function is given enabling for higher rank approximations to better represent the possible local features of the solution. We thus find that the approximation is more accurate as the rank increases. From this example, we can draw the following conclusions:

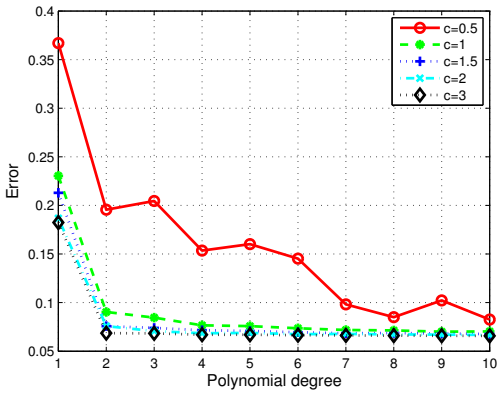
- a heuristic rule to determine the number of samples needed in order to have a stable low rank approximation grows only linearly with dimension $d$ and rank $m$ and is given by $Q = dm(p+1)^2$,
- better solutions are obtained with high rank approximations, provided that enough model evaluations are available.

Quite often in practice, we do not have enough model evaluations and hence we may not be able to achieve good approximations with limited sample size. This is particularly true for certain classes of non smooth functions. One possible solution is a good choice of bases that are sufficiently rich (such as piecewise polynomials or wavelets) and that can capture simultaneously both global and local features of the model function. However, the sample size may not be sufficient enough to obtain good approximations with ordinary least squares in progressive rank one corrections due to large number of basis functions. We illustrate in section 5.2 and 5.3 that, in such cases, performing approximation in *sparse* low rank tensor subsets (i.e.

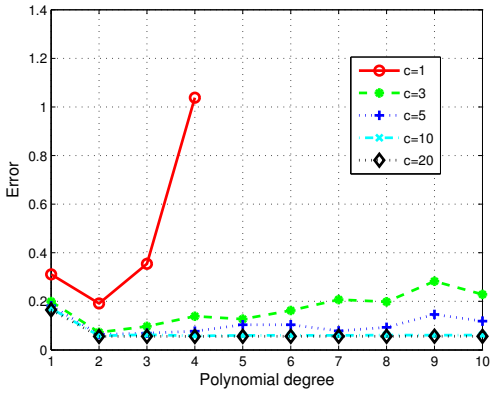
using $\ell_1$ regularization in alternating minimization algorithm) allows more accurate approximation of the model function. In addition, we illustrate in section 5.4 that approximation in sparse low rank tensor subsets leads to a stable approximation with limited number of samples even for high degree polynomial spaces.



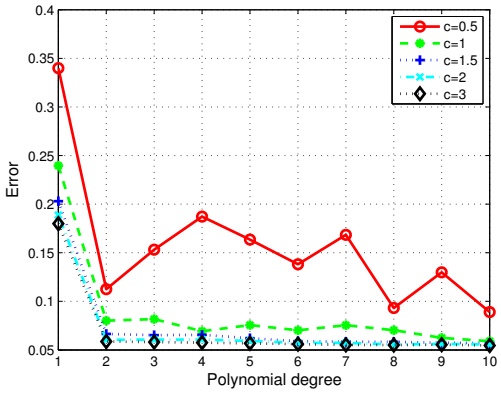

(a) $m = 2$



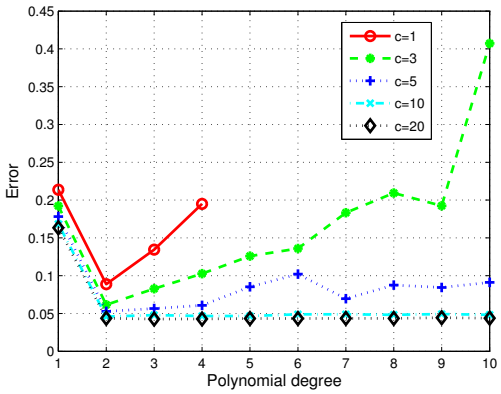

(b) $m = 2$



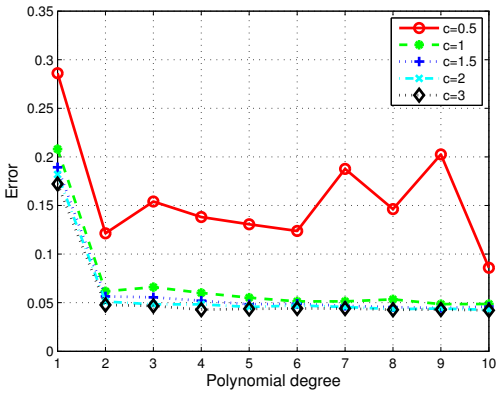

(c) $m = 3$




(d) $m = 3$




(e) $m = 4$




(f) $m = 4$

**Figure 4.2:** Friedman function: Evolution of approximation error $\varepsilon(u_m, u)$ with respect to polynomial degree $p$ with $Q = cd(p+1)$ ((a),(c),(e)) and $Q = cd(p+1)^2$ ((b),(d),(f)) with several values of $c$ and for $m = 2, 3, 4$.

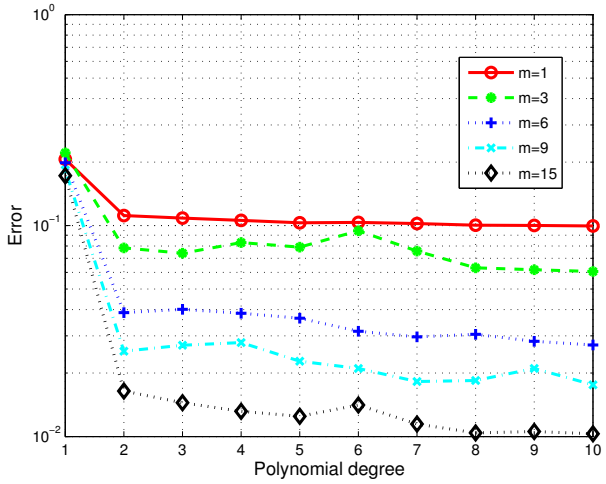



**Figure 4.3:** Friedman function: Evolution of error $\varepsilon(u_m, u)$ with respect to polynomial degree $p$ for different values of $m$ using sample size given by quadratic rule $Q = dm(p+1)^2$.

## 5.2 Analytical model: Checker-board function

### 5.2.1 Function and approximation spaces

We now test Algorithm 8 on the so-called checker-board function $u(\xi_1, \xi_2)$ in dimension $d = 2$ illustrated in figure 4.4. The purpose of this test is to illustrate that, given appropriate bases, in this case piecewise polynomials, Algorithm 8 allows the

detection of sparsity and hence construction of a sequence of optimal sparse rank-$m$ approximations with few samples.
Random variables $\xi_1$ and $\xi_2$ are independent and uniformly distributed on $[0,1]$. The checker-board function is a rank-2 function

$$u(\xi_1,\xi_2)=\sum_{i=1}^{2} w_i^{(1)}(\xi_1)w_i^{(2)}(\xi_2)$$

with $w_1^{(1)}(\xi_1)=c(\xi_1)$, $w_1^{(2)}(\xi_2)=1-c(\xi_2)$, $w_2^{(1)}(\xi_1)=1-c(\xi_1)$ and $w_2^{(2)}(\xi_2)=c(\xi_2)$ where $c(\xi_k)$ is the crenel function defined by:

$$c(\xi_k)=\begin{cases}1 & \text{on } [0,\frac{1}{6}[+2n\frac{1}{6},\ n=0,1,2\\ 0 & \text{on } [\frac{1}{6},\frac{2}{6}[+2n\frac{1}{6},\ n=0,1,2\end{cases}.$$

For approximation spaces $\mathcal{S}_{n_k}^k$, $k\in\{1,2\}$, we introduce piecewise polynomials of de-

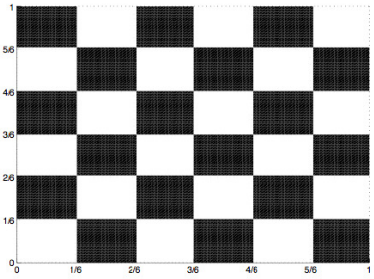



**Figure 4.4:** Checker-board function.

gree $p$ defined on a uniform partition of $\Xi_k$ composed by $s$ intervals, corresponding to $n_k=s(p+1)$. We denote by $\mathcal{S}_{n_k}^k=\mathbb{P}_{p,s}$ the corresponding space (for ex. $\mathbb{P}_{2,3}$ denotes piecewise polynomials of degree 2 defined on the partition $\{(0,\frac{1}{3}),(\frac{1}{3},\frac{2}{3}),(\frac{2}{3},1)\}$). We use an orthonormal basis composed of functions whose supports are one element of the partition and whose restrictions on these supports are rescaled Legendre polynomials.
Note that when using a partition into $s=6n$ intervals, $n\in\mathbb{N}$, then the checker-board function can be exactly represented, that means $u\in\mathbb{P}_{p,6n}\otimes\mathbb{P}_{p,6n}$ for all $p$ and $n$. Also, the solution admits a sparse representation in $\mathbb{P}_{p,6n}\otimes\mathbb{P}_{p,6n}$ since an exact representation is obtained by only using piecewise constant basis functions ($u\in\mathbb{P}_{0,6n}\otimes\mathbb{P}_{0,6n}$). The effective dimensionality of the checker-board function is $2\times 2\times 6=24$, which corresponds to the number of coefficients required for storing the rank-2 representation of the function. We expect our algorithm to detect the low-rank of the function and also to detect its sparsity.

### 5.2.2 Results

Algorithm 8 allows the construction of a sequence of sparse rank-$m$ approximations $u_m$ in $\mathcal{S}^1_{n_1} \otimes \mathcal{S}^2_{n_2}$. We estimate optimal rank-$m_{op}$ using 3-fold cross validation.
In order to illustrate the accuracy of approximations in sparse low rank tensor subsets, we compare the performance of $\ell_1$-regularization within the alternating minimization algorithm (step 4 of Algorithm 8) with no regularization (OLS) and the $\ell_2$-regularization (see Remark 4.1 for the description of these alternatives).
Table 4.1 shows the error $\varepsilon(u_{m_{op}}, u)$ obtained for the selected optimal rank $m_{op}$, without and with updating step of Algorithm 8, and for the different types of regularization during the correction step 4 of Algorithm 8.
The results are presented for a sample size $Q = 200$ and for different function spaces $\mathcal{S}^1_{n_1} = \mathcal{S}^2_{n_2} = \mathbb{P}_{p,s}$. $P$ denotes the dimension of the space $\mathcal{S}^1_{n_1} \otimes \mathcal{S}^2_{n_2}$. We observe that, for $\mathbb{P}_{2,3}$, the solution is not sparse on the corresponding basis and $\ell_1$-regularization does not provide a better solution than $\ell_2$-regularization since the approximation space is not adapted. However, when we choose function spaces that are sufficiently rich for the solution to be sparse, we see that $\ell_1$-regularization within the alternating minimization algorithm outperforms other types of regularization and yields low rank approximations of the function almost at the machine precision. This is because $\ell_1$-regularization is able to select non zero coefficients corresponding to appropriate basis functions of the piecewise polynomial approximation space. For instance, when $\mathbb{P}_{5,6}$ is used as the approximation space, only 3 (out of 36) non zero coefficients corresponding to piecewise constant bases are selected by $\ell_1$ regularization along each dimension in each rank one element (that is the sparsity ratio is $\varrho_1 \approx 0.08$ for each rank-one element), thus yielding an almost exact recovery. We also find that $\ell_1$-regularization allows recovering the exact rank-2 approximation of the function. In fig. 4.5, we plot the sparsity pattern of the functions $v^k_i(\xi_k), 1 \leq i \leq 2, 1 \leq k \leq 2$ using $Q = 200$ samples and approximation space $(\mathbb{P}_{2,6} \otimes \mathbb{P}_{2,6})$. We find that as the alternating least squares iteration increases, non zero coefficients corresponding to pertinent basis (i.e. constant) functions are retained whereas coefficients corresponding to higher order functions are zero.
From this analytical example, several conclusions can be drawn:

- $\ell_1$-regularization in alternating least squares algorithm is able to detect sparsity and hence gives very accurate approximations using few samples as compared to OLS and $\ell_2$-regularizations,

- updating step selects the most pertinent rank-one elements and gives an approximation of the function with a lower effective rank.

## 5.3 Analytical model: Rastrigin function

For certain classes of non smooth functions, wavelet bases form an appropriate choice as they allow the simultaneous description of global and local features [49]. In this

**Table 4.1:** Checker-board function: Relative error $\varepsilon(u_{m_{op}}, u)$ and optimal rank $m_{op}$ estimation of Checker-board function with various regularizations for $Q = 200$ samples. $P$ is the dimension of the approximation space. ('-' indicates that none of the rank-one elements were selected during the update step).

| | Ordinary Least Square | | $\ell_2$ | | $\ell_1$ | |
|---|---|---|---|---|---|---|
| Approximation space | Error | $m_{op}$ | Error | $m_{op}$ | Error | $m_{op}$ |
| $\mathcal{R}_m(\mathbb{P}_{2,3} \otimes \mathbb{P}_{2,3}), P = 9^2$ | 0.527 | 2 | 0.508 | 2 | 0.507 | 2 |
| $\mathcal{R}_m(\mathbb{P}_{2,6} \otimes \mathbb{P}_{2,6}), P = 18^2$ | 0.664 | 2 | 0.061 | 8 | $2.41\,10^{-13}$ | 2 |
| $\mathcal{R}_m(\mathbb{P}_{2,12} \otimes \mathbb{P}_{2,12}), P = 36^2$ | - | - | 0.566 | 4 | $1.1\,10^{-12}$ | 2 |
| $\mathcal{R}_m(\mathbb{P}_{5,6} \otimes \mathbb{P}_{5,6}), P = 36^2$ | - | - | 0.623 | 3 | $7.93\,10^{-13}$ | 2 |
| $\mathcal{R}_m(\mathbb{P}_{10,6} \otimes \mathbb{P}_{10,6}), P = 66^2$ | - | - | 0.855 | 10 | $7.88\,10^{-13}$ | 2 |

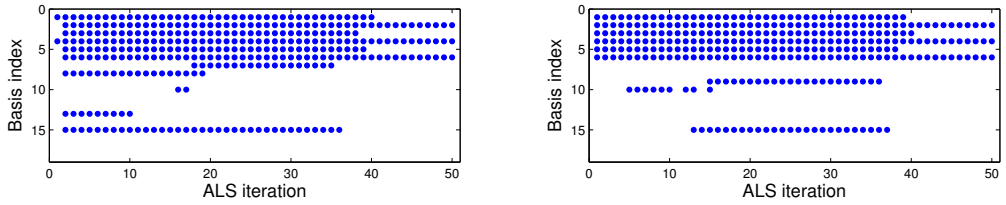
example, we demonstrate the use of our algorithm with polynomial wavelet bases by studying 2-dimensional Rastrigin function. The function is given by

$$u(\xi) = 20 + \sum_{i=1}^{2} (\xi_i^2 - 10\cos(2\pi\xi_i))$$

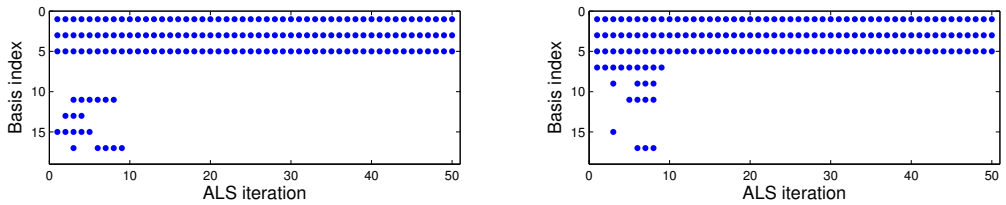
where $\xi_1, \xi_2$ are independent random variables uniformly distributed in [-4,4]. We consider two types of approximation spaces $\mathcal{S}_{n_k}^k$, $k \in \{1, 2\}$:

- spaces of polynomials of degree 7, using Legendre polynomial chaos basis, denoted $\mathbb{P}_7$,
- spaces of polynomial wavelets with degree 4 and resolution level 3, denoted $\mathbb{W}_{4,3}$.

We compute a sequence of sparse rank-$m$ approximations $u_m$ in $\mathcal{S}_{n_1}^1 \otimes \mathcal{S}_{n_2}^2$ using Algorithm 8 and an optimal rank approximation $u_{m_{op}}$ is selected using 3-fold cross validation. Figure 4.6(a) shows the convergence of this optimal approximation with respect to the sample size $Q$ for the two different approximation spaces. We find that the solution obtained with classical polynomial basis functions is inaccurate and does not improve with increase in sample size. Thus, polynomial basis functions are not a good choice to obtain a reasonably accurate estimate. On the other hand, when we use wavelet approximation bases, the approximation error reduces progressively with increase in sample size. Figure 4.6(b) shows the convergence of the optimal wavelet approximation with respect to the sample size $Q$ for different regularizations within the alternating minimization algorithm of the correction step. The $\ell_1$ regularization is more accurate when compared to both OLS and $\ell_2$ regularization, particularly for few model evaluations. We can thus conclude that a good choice of basis functions is important in order to fully realize the potential of sparse $\ell_1$ regularization in the tensor approximation algorithm.




(a) $v_1^1$ (b) $v_1^2$

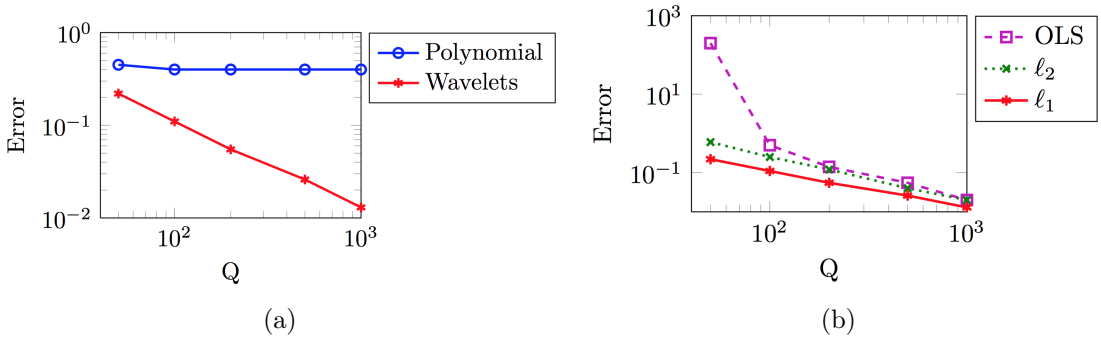



(c) $v_2^1$ (d) $v_2^2$

**Figure 4.5:** Checker-board function: Evolution of sparsity pattern of functions $v_i^k(\xi_k), 1 \leq i \leq 2, 1 \leq k \leq 2$ w.r.t ALS iterations using $Q = 200$ samples in $(\mathbb{P}_{2,6} \otimes \mathbb{P}_{2,6})$

Figure 4.7 shows the convergence of the approximation obtained with Algorithm 8 using different sample sizes. We find that as the sample size increases, we get better approximations with increasing rank. Conversely, if only very few samples are available, then a very low rank approximation, even rank one, is able to capture the global features. The proposed method provides the simplest representation of the function with respect to the available information.

We finally analyze the robustness of Algorithm 8 with respect to the sample sets. We use wavelet bases. An optimal rank approximation $u_{m_{op}}$ is selected using 3-fold cross validation. We compare this algorithm with a direct sparse least-squares approximation in the tensorized polynomial wavelet space (no low-rank approximation), using $\ell_1$-regularization (use of Algorithm 5). Figure 4.8 shows the evolution of the relative error with respect to the sample size $Q$ for these two




(a) (b)

**Figure 4.6:** Rastrigin function: Evolution of error $\varepsilon(u_{m_{op}}, u)$ with respect to the number of samples $Q$. Approximations obtained with Algorithm 8 with optimal rank selection: (a) for the two different approximation spaces $\mathbb{P}_7 \otimes \mathbb{P}_7$ ($P = 64$) and $\mathbb{W}_{4,3} \otimes \mathbb{W}_{4,3}$ ($P = 1600$) and (b) for different types of regularizations with approximation space $\mathbb{W}_{4,3} \otimes \mathbb{W}_{4,3}$.

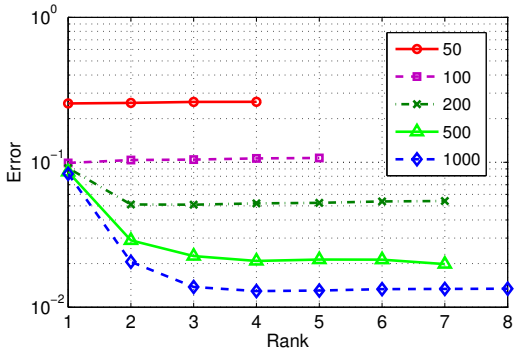



**Figure 4.7:** Rastrigin function: Evolution of error $\varepsilon(u_m, u)$ of approximations obtained using Algorithm 8 with respect to rank $m$ for different sample sizes on wavelet approximation space $\mathbb{W}_{4,3} \otimes \mathbb{W}_{4,3}$ ($P = 1600$)

strategies. The vertical lines represent the scattering of the error when different sample sets are used. We observe a smaller variance of the obtained approximations when exploiting low-rank representations. This can be explained by the lower dimensionality of the representation, which is better estimated with a few number of samples. On this simple example, we see the interest of using greedy constructions of sparse low-rank representations when only a small number of samples is available, indeed the problem is reduced to one where elements of subsets of small dimension are to be estimated. The interest of using low-rank representations should also become clear when dealing with higher dimensional problems.

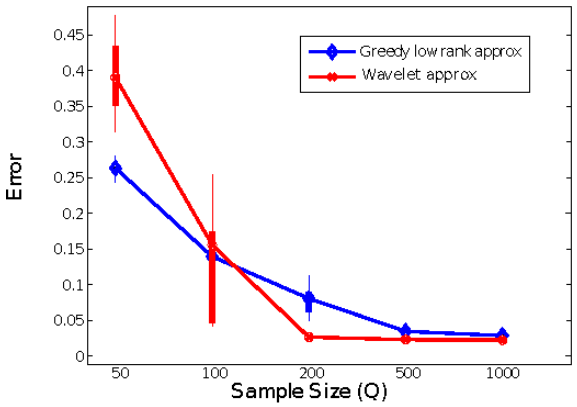
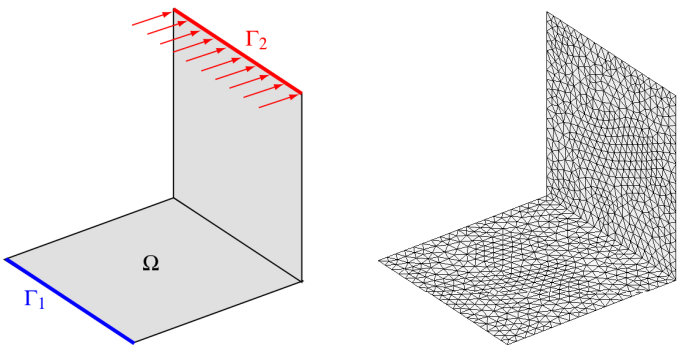



**Figure 4.8:** Rastrigin function: Evolution of error $\varepsilon(u_{m_{op}}, u)$ with respect to sample size $Q$. (Red line) approximation obtained with direct least-squares approximation with $\ell_1$-regularization on the full polynomial wavelet approximation space $\mathbb{W}_{4,3} \otimes \mathbb{W}_{4,3}$, (blue line) approximation obtained with Algorithm 8 (with $\ell_1$-regularization) and with optimal rank selection.

## 5.4 A model problem in structural vibration analysis

### 5.4.1 Model problem

We consider a forced vibration problem of a slightly damped random linear elastic structure. The structure composed of two plates is clamped on part $\Gamma_1$ of the boundary and submitted to a harmonic load on part $\Gamma_2$ of the boundary as represented in figure 4.9(a). A finite element approximation is introduced at the spatial level using a mesh composed of 1778 DKT plate elements (see figure 4.9(b)) and leading to a discrete deterministic model with $N = 5556$ degrees of freedom. The resulting




(a) (b)

**Figure 4.9:** Elastic plate structure under harmonic bending load. Geometry and boundary conditions (a) and finite element mesh (b).

discrete problem at frequency $\omega$ writes

$$(-\omega^2\mathbf{M} + i\omega\mathbf{C} + \mathbf{K})\mathbf{u} = \mathbf{f},$$

where $\mathbf{u} \in \mathbb{C}^N$ is the vector of coefficients of the approximation of the displacement field and $\mathbf{M}$, $\mathbf{K} = E\tilde{\mathbf{K}}$ and $\mathbf{C} = i\omega\eta E\tilde{\mathbf{K}}$ are the mass, stiffness and damping matrices respectively. The non-dimensional analysis considers a unitary mass density and a circular frequency $\omega = 0.67$ rad.s$^{-1}$. The Young modulus $E$ and the damping parameter $\eta$ are defined by

$$E = \begin{cases} 0.975 + 0.025\xi_1 & \text{on horizontal plate,} \\ 0.975 + 0.025\xi_2 & \text{on vertical plate,} \end{cases}$$

$$\eta = \begin{cases} 0.0075 + 0.0025\xi_3 & \text{on horizontal plate,} \\ 0.0075 + 0.0025\xi_4 & \text{on vertical plate,} \end{cases}$$

where the $\xi_k \sim U(-1, 1)$, $k = 1, \cdots, 4$, are independent uniform random variables (here $d = 4$). We define the quantity of interest

$$I(u)(\xi) = \log \|u_c\|,$$

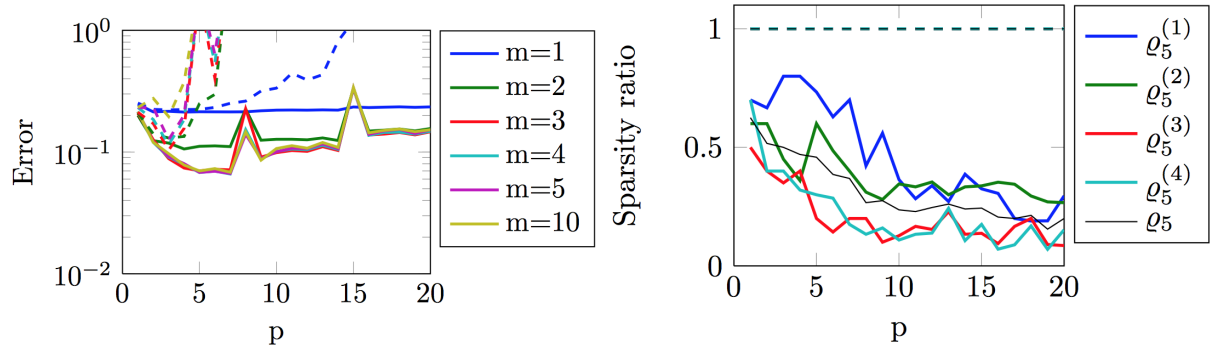
where $u_c$ is the displacement of the top right node of the two plates structure.

### 5.4.2 Impact of regularization and stochastic polynomial degree

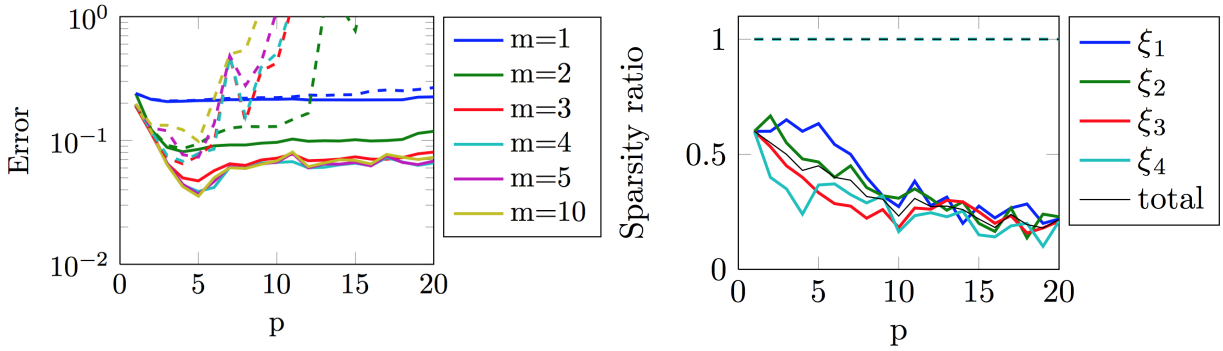
In this example, we illustrate that the approximation in sparse low rank tensor subsets is robust when increasing the degree of underlying polynomial approximation spaces with a fixed number of samples $Q$. We use Legendre polynomial basis functions with degree $p = 1$ to 20 and denote by $\mathbb{P}_p$ the corresponding space of polynomials of maximal degree $p$ in each dimension. A rank-$m$ approximation is searched in the isotropic polynomial space $\mathbb{P}_p \otimes \cdots \otimes \mathbb{P}_p$. Figure 4.10(left column) shows the error $\varepsilon(I_m, I)$ as a function of the polynomial degree $p$ for different ranks $m$ for three different sizes of the sample set, $Q = 80, 200$ and 500. The low rank approximation $I_m$ is computed with and without sparsity constraint, *i.e.* we compare OLS (dashed lines) and $\ell_1$ regularization (solid lines) in correction step 4 of Algorithm 8. Figure 4.11 summarizes the error $\varepsilon(I_{m_{op}}, I)$ for different sizes of sample sets for the optimal rank $m_{op}$ approximation when using $\ell_1$-regularization (solid lines) and using OLS (dashed lines). We find that OLS yields a deterioration with higher polynomial order. This is consistent with the conclusions in section 5.1 and the quadratic rule according to which convergence is observed for $Q \geq dm(p+1)^2$ and a deterioration is expected otherwise. On the other hand, we see that $\ell_1$-regularization gives a more stable approximation with increasing polynomial order and also gives a more accurate approximation than the approximation obtained with OLS. This can be attributed to the selection of pertinent basis functions obtained by imposing sparsity constraint. Indeed, we

clearly see in figure 4.10(right column) that the sparsity ratio $\rho_5$ for sparse low rank approximation (solid black line) decreases with increasing polynomial degree (see 5 for definition of sparsity ratio). Along with the total sparsity ratio, the partial sparsity ratios $\varrho_5^{(k)}$ in each dimension $k = 1$ to 4 are plotted in figure 4.10(right column). We see that $\ell_1$-regularization exploits sparsity especially in dimensions 3 and 4 corresponding to the damping coefficients, indeed the quantity of interest has smooth dependance on variables $\xi_3$ and $\xi_4$ whereas it has a high non linear behavior with respect to $\xi_1$ and $\xi_2$. Figure 4.12 shows the reference quantity of interest and the rank-$m_{op}$ approximation $I_{m_{op}}$ obtained using $\ell_1$-regularization and polynomial degree 3 constructed from $Q = 200$ samples.

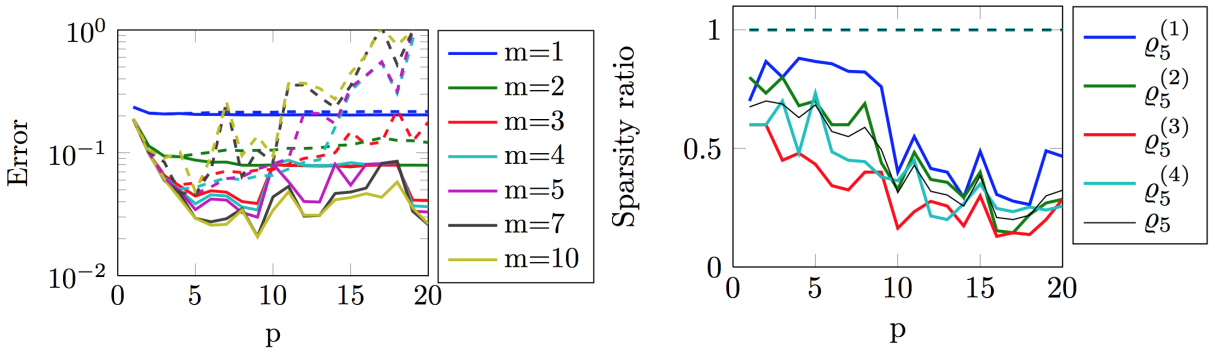
This illustration also points out that when a small number of model evaluations are available, for instance $Q = 80$, the method is able to capture correctly global features of the function and a low rank approximation ($m = 3$) is selected as the best approximation regarding the available information. As the number of samples increases, higher rank approximation are selected that capture the local features of the function more accurately.




(a) $Q = 80$




(b) $Q = 200$




(c) $Q = 500$

**Figure 4.10:** Two plate structure: Left column: evolution of approximation error $\varepsilon(I_m, I)$ with respect to polynomial degree $p$ with a fixed number of samples $Q = 80, Q = 200$ and $Q = 500$. Polynomial approximations obtained using $\ell_1$-regularization (solid lines) and using OLS (dashed lines). Right column: total and partial sparsity ratios with respect to polynomial degree for $m = 5$.

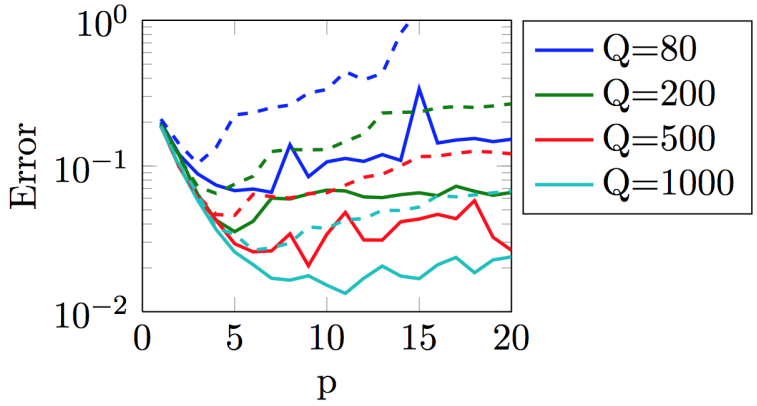
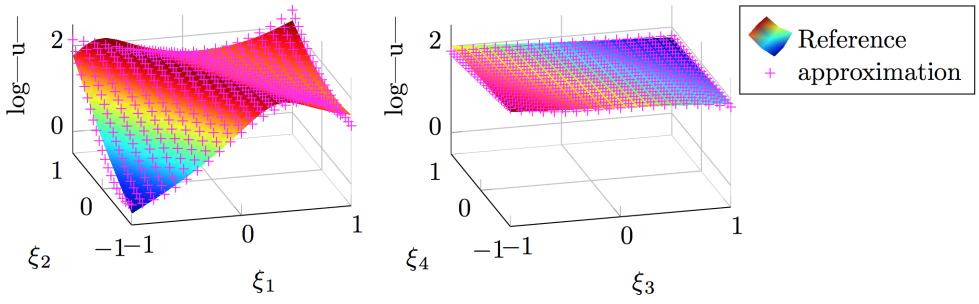



**Figure 4.11:** Two plate structure: Evolution of approximation error $\varepsilon(I_{m_{op}}, I)$ with respect to polynomial degree $p$ for different sizes of sample sets (random sampling). Polynomial approximations obtained using $\ell_1$-regularization (solid lines) and using OLS (dashed lines).




(a) $\xi_3 = 0, \xi_4 = 0$ (b) $\xi_1 = 0, \xi_2 = 0$

**Figure 4.12:** (Two plate structure) Response surface : (surface) reference and (dotted) rank-$m_{op}$ approximation obtained with polynomial basis $p = 3$, $Q = 200$ using $l_1$ regularization.

## 5.5 A stochastic partial differential equation

In this example, we consider a stationary advection diffusion reaction equation on a spatial domain $\Omega = (0, 1)^2$ (Fig. 4.13) where the source of uncertainty comes from the diffusion coefficient which is a random field. The problem is:

$$\begin{aligned} -\nabla\cdot(\mu(x,\xi)\nabla u) + c\cdot\nabla u + \kappa u &= f \text{ on } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned}$$

where $\kappa = 10$ is a deterministic reaction coefficient and $c = 250(x - \frac{1}{2}, \frac{1}{2} - y)$ is a deterministic advection velocity. The source term is deterministic and is defined by $f = 100 I_{\Omega_1}$, where $\Omega_1 = (0.7, 0.8) \times (0.7, 0.8) \subset \Omega$, where $I_{\Omega_1}$ is the indicator

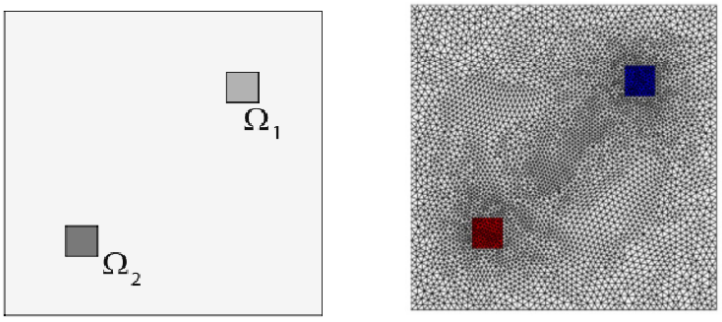



**Figure 4.13:** Advection diffusion reaction problem: Domain and finite element mesh

function of $\Omega_1$. $\mu(x,\xi)$ is a random field defined by

$$\mu(x,\xi) = \mu_0 + \sum_{k=1}^{100} \sqrt{\sigma_k}\mu_k(x)\xi_k \tag{4.10}$$

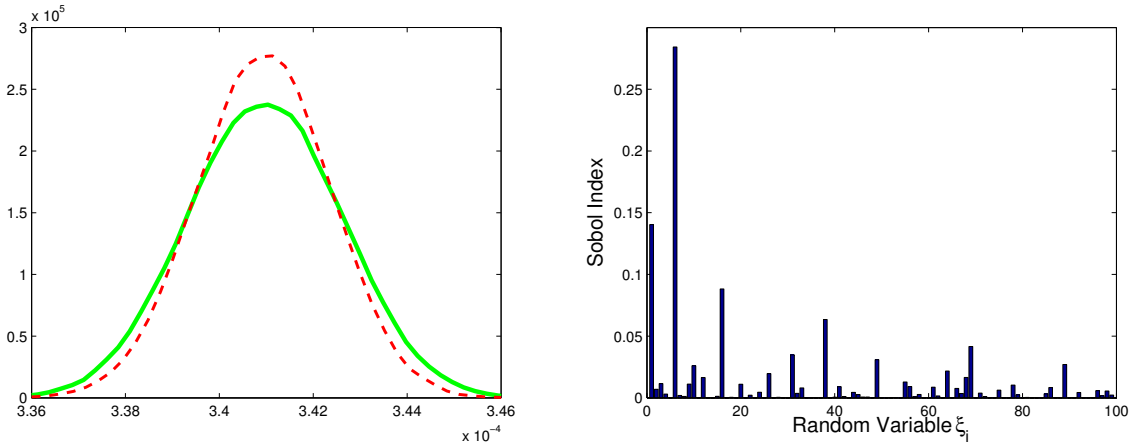
where $\mu_0 = 1$ is the mean value of $\mu$, where the $\xi_k \sim U(-1,1)$ are mutually independent uniform random variables and where the $\mu_k$ are a set of $L^2(\Omega)$-orthonormal spatial functions.
The couples $(\mu_k, \sigma_k) \in L^2(\Omega) \times \mathbb{R}^+$ are chosen as the 100 dominant eigenpairs of eigenproblem $T(\mu_k) = \sigma_k\mu_k$, where $T$ is the kernel operator

$$T : v \in L^2(\Omega) \mapsto \int_\Omega \alpha(x,y)v(y)dy \in L^2(\Omega)$$

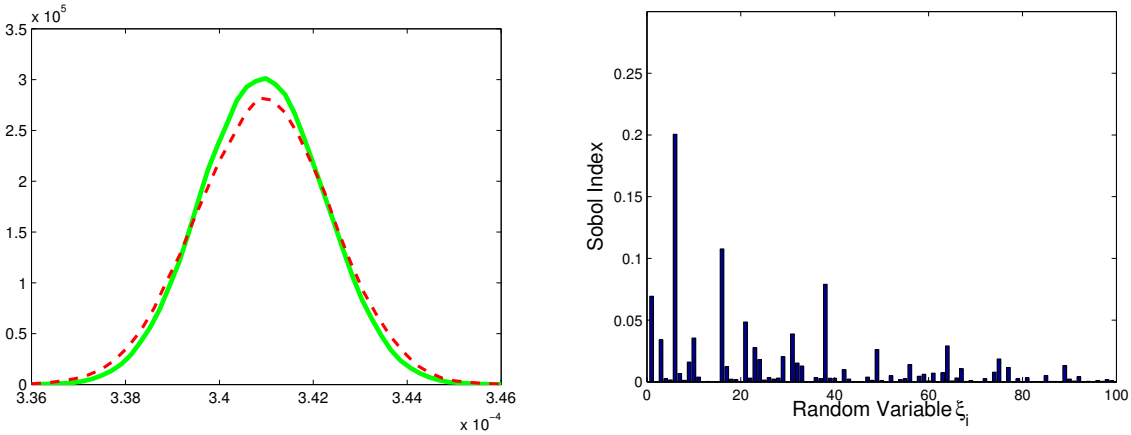
with $\alpha(x,y) = 0.2^2 exp(-\frac{\|x-y\|^2}{l_c^2})$ with $l_c$ the correlation length. The equation (4.10) then corresponds to truncated version of a homogeneous random field with mean 1, standard deviation $\frac{0.2}{\sqrt{3}}$ and exponential square covariance function with correlation length $l_c$. The $d = 100$ random parameters $\xi = (\xi_k)_{k=1}^d$ define a probability space $(\Xi, \mathcal{B}, P_\xi)$, with $\Xi = (-1,1)^d$ and $P_{\boldsymbol{\xi}}$ the uniform probability measure. We introduce approximation spaces $\mathcal{S}_{n_k}^k = \mathbb{P}_p(-1,1)$ which are spaces of polynomials with degree $p = 3$.

For this example, we plot the probability density function and first order Sobol sensitivity indices for different sample sizes in figure 4.14. We find that the proposed method is very effective in this very high dimensional stochastic problems. The number of model evaluations required to obtain very accurate solution is as low as $3d(p+1)$. This is orders of magnitude less than a classical PC approximation (which would require $Q = (p+d)!/(p!d!)$ model evaluations. For example in this problem for $p = 3$, we have $Q = 176\,851$). Note that the efficiency of the proposed tensor approximation methods on this particular example is due to the effective low rank of the solution and sparse polynomial approximation method [9] also gives

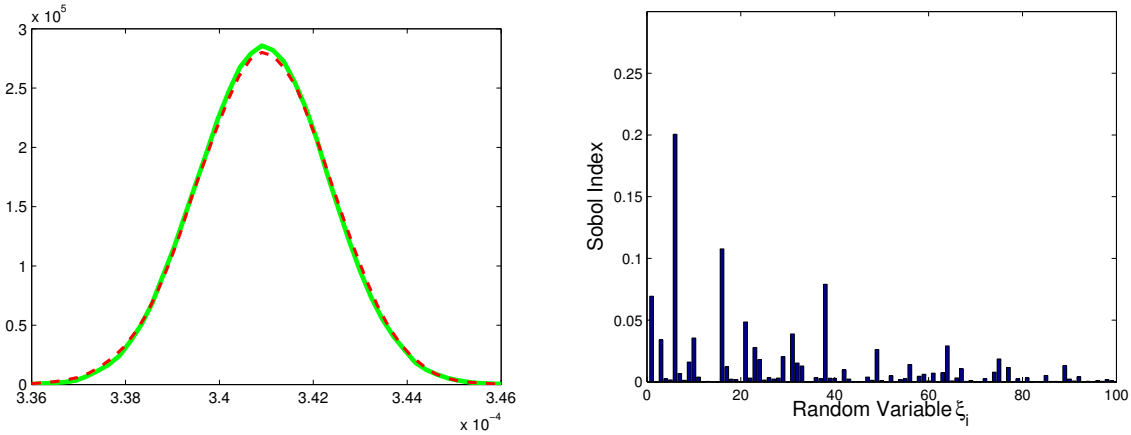
similar accuracy for the same sample size. We also illustrate that post-processing in low rank canonical format is quite straightforward and computationally very cheap and even with very few sample, we can obtain coarse but quite good ordering of Sobol indices.




(a) $Q = 100$




(b) $Q = 200$




(c) $Q = 1000$

**Figure 4.14:** Advection diffusion reaction problem: Left column: evolution of pdf (reference dashed line) and right column: Sobol sensitivity indices $S_k, 1 \leq k \leq 100$, for number of samples $(a)Q = 100, (b)Q = 200$ and $(c)Q = 1000$.

# 6 Extension to vector valued functions: Diffusion equation with multiple inclusions

In this section we present a simple way to extend these algorithms to vector valued functions using Karhunen Loeve expansion. We consider a stationary diffusion problem defined on a two dimensional domain $\Omega = (0,1) \times (0,1)$ (see figure 4.15):

$$-\nabla \cdot (\kappa \nabla u) = I_D(x) \quad \text{on} \quad \Omega, \quad u = 0 \quad \text{on} \quad \partial\Omega,$$

where $D = (0.4, 0.6) \times (0.4, 0.6) \subset \Omega$ is a square domain and $I_D$ is the indicator function of $D$. The diffusion coefficient is defined by

$$\kappa = \begin{cases} \xi_k & \text{on } C_k,\ 1 \leq k \leq 8, \\ 1 & \text{on } \Omega \backslash (\cup_{k=1}^{8} C_k) \end{cases}$$

where the $C_k$ are circular domains and where the $\xi_k \sim U(0.01, 1.0)$ are independent uniform random variables. We define the quantity of interest

$$I(u)(\xi) \ = \ \int_D u(x, \xi) dx.$$

We wish to obtain a reduced model for the random field $u(x, \xi) \in \mathcal{V}_N \otimes L^2_{P_{\xi_1}}(\Xi_1) \otimes$

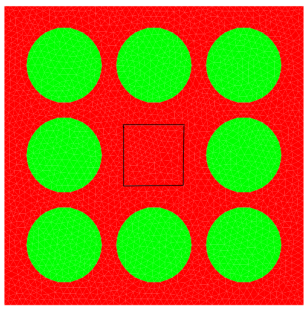


**Figure 4.15:** Diffusion problem with multiple inclusions.

$\ldots \otimes L^2_{P_{\xi_d}}(\Xi_d)$, where $\mathcal{V}_N$ is a $N$-dimensional finite element approximation space used for the discretization of the partial differential equation. A straightforward application of the previous methodology would require to evaluate $u$ at certain number of realizations of the input random vector $\xi$ and to perform least-squares approximation for each node of the finite element mesh. However, this may not be feasible for large number of degrees of freedom $N$. Hence we wish to obtain a reasonably accurate low rank representation of the random field $u(x, \xi)$. For that purpose, we apply an empirical Karhunen-Loeve decomposition. Let $\mathbf{u}(y^q) \in \mathbb{R}^N$, $q \in \{1 \ldots Q\}$, represent the finite element solutions (nodal values) associated with the $Q$ realizations of the input random vector $\xi$. We denote by $\mathbf{u}_0 = \frac{1}{Q} \sum_{q=1}^{Q} \mathbf{u}(y^q)$ the empirical

mean, and by $\widetilde{\mathbf{u}}(y^q) = \mathbf{u}(y^q) - \mathbf{u}_0$. We gather the centered realizations in a matrix $\widetilde{\boldsymbol{U}} = (\widetilde{\mathbf{u}}(y^1) \dots \widetilde{\mathbf{u}}(y^Q))$. We compute the Singular Value Decomposition (SVD) $\widetilde{\boldsymbol{U}} = \sum_{i=1}^{N} \sigma_i \mathbf{v}_i \otimes \mathbf{z}_i$, where the $\mathbf{v}_i \in \mathbb{R}^N$ and the $\mathbf{z}_i \in \mathbb{R}^Q$ are respectively the left and right singular vectors of $\widetilde{\boldsymbol{U}}$, and the $\sigma_i$ are the corresponding singular values ordered with decreasing values. Then, this decomposition is truncated by retaining only the $N^\star$ dominant singular values: $\widetilde{\boldsymbol{U}} \approx \sum_{i=1}^{N^\star} \sigma_i \mathbf{v}_i \otimes \mathbf{z}_i$. It corresponds to the following approximation of the finite element solution (truncated empirical Karhunen-Loeve decomposition):

$$\mathbf{u}(\xi) \approx \mathbf{u}_0 + \sum_{i=1}^{N^\star} \sigma_i \mathbf{v}_i z_i(\xi) \tag{4.11}$$

where the $z_i(\xi)$ are random variables whose evaluations at samples $\{y^q\}_{q=1}^{Q}$ are gathered in the vectors $\mathbf{z}_i$. This procedure reduces the effective dimensionality of the solution field, which is now a function of only $N^\star$ random variables. Our least-squares-based tensor approximation method can be applied to each random variable $z_i$ separately. We perform the above procedure for multi-inclusion problem and compare the solution obtained for sample sets of three different sizes ($Q = 100, 500, 1500$). Figure 4.16 shows the relative error v/s the rank of the SVD decompostion of $\widetilde{\boldsymbol{U}}$ (with respect to Frobenius norm). As can be seen, the error drops significantly after 8 modes. Thus, we choose $N^\star = 8$ and we apply the least-squares-based tensor approximation method for the random variables $\{z_i(\xi)\}_{i=1}^{8}$, using Algorithm 8 for each input vector $\mathbf{z}_i$ independently. We consider the finest tensor structure $\mathcal{S}^1 \otimes \dots \otimes \mathcal{S}^8$ with polynomial spaces of total degree $p = 5$. Figure 4.17 shows the cross validation

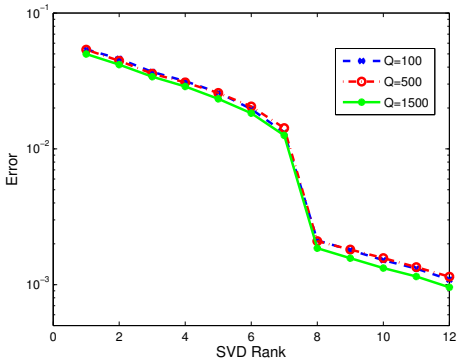



**Figure 4.16:** Diffusion problem: Error of the truncated SVD with respect to the rank for different sample size.

error of the low rank approximations of $\{z_i(\xi)\}_{i=1}^{8}$ with different sample sizes. As can be expected, the cross validation error reduces with increase in sample size and we can select the best low-rank approximation (with optimal rank selection) for each mode to construct an approximation of $\mathbf{u}(\xi)$ under the form (4.11). Let $\bar{I}$ denote

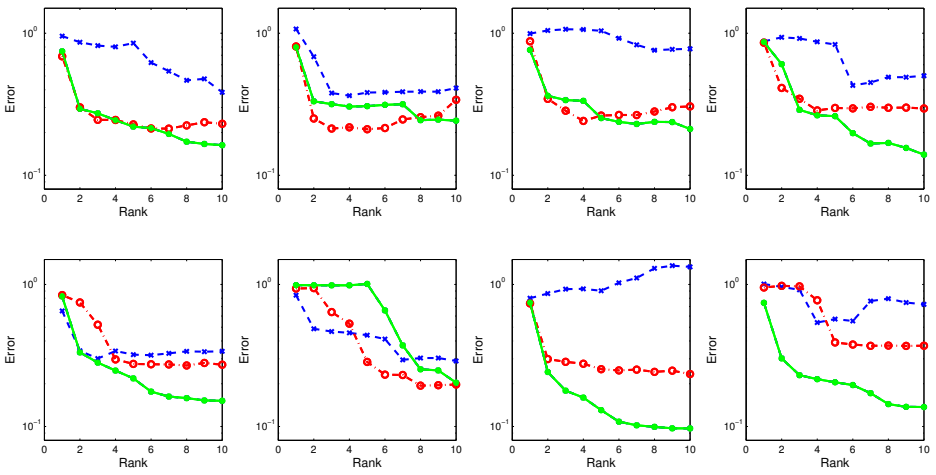



**Figure 4.17:** Diffusion problem: Evolution of the 3-fold cross validation error of the low rank approximation of the first 8 random variables $\{z_i(\xi)\}_{i=1}^{8}$ with the rank (left to right) for sample size 100 (blue), 500 (red) and 1500 (green).

the approximation of quantity of interest $I$ obtained by the post treatment of the low rank approximation of $\mathbf{u}(\xi)$. Figure 4.18 shows the error on the mean value of the quantity of interest obtained with the reduced model for different sample sizes. The same figure also plots the error on the empirical mean of the quantity of interest. We observe that, for few samples (i.e. 100), the mean value estimated from reduced model is not good since the corresponding approximation of the stochastic modes are inaccurate. However, as we increase the sample size, this strategy gives better approximation of stochastic modes and hence more accurate estimation of the quantity of interest.

# 7 Conclusion

In this chapter, we detailed the implementation of sparse low-rank canonical tensor approximation method for multivariate stochastic functions. Greedy algorithms for low-rank tensor approximation have been combined with sparse least-squares approximation methods in order to obtain a robust construction of sparse low-rank tensor approximations in high dimensional approximation spaces. The ability of the proposed method to detect and exploit low-rank and sparsity was illustrated on analytical models and on a stochastic partial differential equations. We also presented a simple method to extend this technique for approximating vector valued functions.

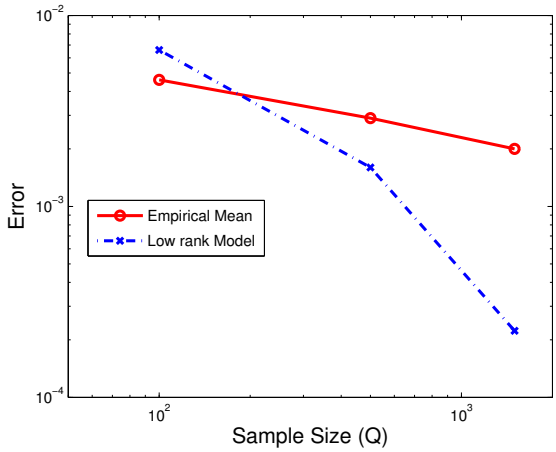



**Figure 4.18:** Diffusion problem: Evolution of the error on the mean value using low rank model and the error on the empirical mean of the quantity of interest with the sample size $Q$.

# Chapter 5

# Approximation in Sparse Tensor Train Format

*In this chapter, we describe the implementation of approximation in sparse tensor train format. In section 1, we introduce sparse low rank tensor train format and its parametrization. Alternating Least-Squares (ALS) algorithm to perform approximation in sparse low rank tensor train format is presented in section 2. Due to the issue of selecting the optimal TT rank in ALS, we introduce ALS with internal rank adaptation in 3.1 and a modified version of ALS (DMRG algorithm) in section 3.2. Finally we illustrate the performance of these algorithms on analytical and numerical examples in section 4.*

## Contents

Approximation of high dimensional stochastic functions in sparse canonical tensor subsets gives the simplest sparse low rank tensor representation of a multivariate function (see chapter 4). Although a canonical tensor based approximation is a good candidate, it suffers from several drawbacks. The optimal canonical rank is not known in advance and approximation with a fixed canonical rank can be ill-posed, the set being closed only if the rank is 1 or the dimension $d$ is 2 [25]. Numerical algorithms for computing an approximate representation in such cases may not yield stable approximations unless using greedy procedure that usually yields suboptimal approximations. It is thus imperative to look for alternative tensor approximation formats, which may have more number of parameters but are well suited for numerical approximations.

The Tensor Train (TT) format has been introduced by Oseledets [61, 59, 60] which offers one of the simplest type of representation of a tensor in tree based formats. Algorithms based on alternating minimization and modified alternating minimization (proposed earlier as density-matrix renormalization group method (DMRG) for simulating quantum systems [80, 65]) for approximation of high-dimensional tensors in TT format have also been introduced. Here, we propose algorithms that approximate high-dimensional functions in sparse TT subsets using very few function evaluations, thus exploiting both low-rank and sparsity structure of the function. In the following section, we introduce the sparse low rank tensor train subset.

# 1 Sparse tensor train format

We introduce the set of multi-indices $\mathcal{I} = \{i = (i_0, i_1, \ldots, i_{d-1}, i_d);\ i_k \in \{1, \ldots, r_k\}\}$ with $r_0 = r_d = 1$. The set of tensors $\mathcal{TT}_r$ in $\mathcal{S}_{\boldsymbol{n}} = \mathcal{S}^1_{n_1} \otimes \ldots \otimes \mathcal{S}^d_{n_d}$ is defined by

$$\mathcal{TT}_r = \left\{ v = \sum_{i \in \mathcal{I}} \bigotimes_{k=1}^{d} v^k_{i_{k-1} i_k};\ v^k_{i_{k-1} i_k} \in \mathcal{S}^k_{n_k} \right\},$$

or equivalently

$$\mathcal{TT}_r = \left\{ v(y) = \sum_{i \in \mathcal{I}} \prod_{k=1}^{d} \langle \boldsymbol{\phi}^k(y_k), \mathbf{v}^k_{i_{k-1} i_k} \rangle ; \mathbf{v}^k_{i_{k-1} i_k} \in \mathbb{R}^{n_k} \right\},$$

where $\mathbf{v}^k_{i_{k-1} i_k}$ is the vector of coefficients of $v^k_{i_{k-1} i_k}$ in the basis $\boldsymbol{\phi}^k = (\phi^k_1, \ldots, \phi^k_{n_k})^T$ of $\mathcal{S}^k_{n_k}$.
The coefficient vectors of $r_{k-1} \times r_k$ functions $v^k_{i_{k-1} i_k} \in \mathcal{S}^k_{n_k}$ can be gathered in 3-order tensor $\mathbf{v}^k \in \mathcal{P}^k = \mathbb{R}^{r_{k-1} \times r_k \times n_k}$ which parametrizes $v^k \in \left(\mathcal{S}^k_{n_k}\right)^{r_{k-1} \times r_k}$. Thus, the subset $\mathcal{TT}_r$ can then be parametrized by the map $F_{\mathcal{TT}_r} : \mathcal{P} \to \mathcal{S}_n$ such that

$\mathcal{P} = \mathcal{P}^1 \times \ldots \times \mathcal{P}^d$ and

$$F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)(y) = \sum_{i \in \mathcal{I}} \bigotimes_k \langle \mathbf{v}^k_{i_{k-1} i_k}, \boldsymbol{\phi}^k(y_k) \rangle \in \mathcal{TT}_r(\mathcal{S}_n).$$

The tensors $\mathbf{v}^k$ are also called the *cores* of the tensor train tensor. Approximation in $\mathcal{TT}_r$ using classical least squares methods possibly enables to recover a good approximation of the solution using a reduced number of samples. However, the samples may not be sufficient in the case where the approximation spaces $\mathcal{S}^k_{n_k}$ have high dimensions $n_k$ or if the components of $r$ are too high, thus resulting in a parametrization of $\mathcal{TT}_r$ with high dimension $\sum_{k=1}^d r_{k-1} r_k n_k$. This difficulty may be circumvented by introducing approximations in a $\boldsymbol{m}$-sparse tensor train subset defined as

$$\mathcal{TT}_r^{\boldsymbol{m}\text{-sparse}} = \left\{ v = F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d),\ \|\mathcal{V}(\mathbf{v}^k)\|_0 \leq m_k \right\},$$

where $\mathcal{V}(\mathbf{v}^k) : \mathbf{v}^k \in \mathcal{P}^k \mapsto \mathcal{V}(\mathbf{v}^k) \in \mathbb{R}^{(r_k r_{k-1} n_k)}$ is the vector obtained by stacking the coefficient vectors $\mathbf{v}^k_{i_{k-1} i_k}$ in a column vector. The effective dimension is thus $\sum_{k=1}^d r_{k-1} r_k m_k \leq \sum_{k=1}^d r_{k-1} r_k n_k$. Due to combinatorial nature of optimization problem formulated in $\mathcal{TT}_r^{\boldsymbol{m}\text{-sparse}}$, performing least squares approximation in this set may not be computationally tractable. We thus introduce a convex relaxation of the $\ell_0$-norm to define the subset $\mathcal{TT}_r^{\gamma}$ as

$$\mathcal{TT}_r^{\gamma} = \left\{ v = F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d),\ \|\mathcal{V}(\mathbf{v}^k)\|_1 \leq \gamma_k \right\}.$$

Performing approximation in this set using least squares becomes computationally tractable (see chapter 3 for a general introduction to sparse low rank formats). In the following, we propose algorithms for the construction of approximations in tensor subset $\mathcal{TT}_r^{\gamma}$, with selection of $\gamma$.

## 2 Construction of sparse tensor train approximation using alternating least square

We determine tensor train approximation $v \in \mathcal{TT}_r^{\gamma}(\mathcal{S})$ of $u$ by solving the least square problem

$$\min_{v \in \mathcal{TT}_r^{\gamma}} \|u - v\|_Q^2 = \min_{\substack{\mathbf{v}^1 \in \mathcal{P}^1, \ldots, \mathbf{v}^d \in \mathcal{P}^d \\ \|\mathcal{V}(\mathbf{v}^1)\|_1 \leq \gamma_1, \ldots, \|\mathcal{V}(\mathbf{v}^d)\|_1 \leq \gamma_d}} \|u - F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)\|_Q^2. \tag{5.1}$$

Problem 5.1 is solved using ALS by successively computing the core tensor $\mathbf{v}^k \in \mathcal{P}^k$ for fixed values of $\mathbf{v}^j, j \neq k$. Thus, for $k = 1, \ldots, d$, we solve the least squares problem given by

$$\min_{\substack{\mathbf{v}^k \in \mathcal{P}^k \\ \|\mathcal{V}(\mathbf{v}^k)\|_1 \leq \gamma_k}} \frac{1}{Q} \sum_{q=1}^{Q} \left( u(y_q) - \langle \mathbf{v}^k, \alpha^k_q \otimes \beta^k_q \otimes \boldsymbol{\phi}^k(y^k_q) \rangle \right)^2. \tag{5.2}$$

Here $\langle x, y \otimes z \otimes t \rangle = \sum_{i=1}^{r_{k-1}} \sum_{j=1}^{r_k} \sum_{l=1}^{n_k} x_{ijl} y_i z_j t_l$ for all $x \in \mathbb{R}^{r_{k-1} \times r_k \times n_k}, y \in \mathbb{R}^{r_k}$, $z \in \mathbb{R}^{r_k}$ and $t \in \mathbb{R}^{n_k}$. $\alpha_q^k = (\alpha_{q,i_{k-1}}^k)_{i_{k-1}=1}^{r_{k-1}}$ is such that $\alpha_{q,i_{k-1}}^k$ is a scalar given by

$$\alpha_{q,i_{k-1}}^k = \sum_{i_1=1}^{r_1} \cdots \sum_{i_{k-2}=1}^{r_{k-2}} \prod_{\mu=1}^{k-1} v_{i_{\mu-1} i_\mu}^k(y_q) \tag{5.3}$$

and $\beta_q^k = (\beta_{q,i_k}^k)_{i_k=1}^{r_k}$ is such that

$$\beta_{q,i_k}^k = \sum_{i_{k+1}=1}^{r_{k+1}} \cdots \sum_{i_d=1}^{r_d} \prod_{\mu=k+1}^{d} v_{i_{\mu-1} i_\mu}^k(y_q). \tag{5.4}$$

$\alpha_q^k$ is a vector obtained from the cores to the left of core on which minimization is performed (i.e. the cores $\mathbf{v}^j, j = 1, \ldots, k-1$). Similarly $\beta_q^k$ is a vector obtained from the cores to the right of the core on which minimization is performed (i.e. the cores $\mathbf{v}^j, j = k+1, \ldots, d$).
Problem 5.2 can be equivalently written in a lagrangian form

$$\min_{\mathbf{v}^k \in \mathcal{P}^k} \|\mathbf{z} - \mathbf{\Psi^k} \mathcal{V}(\mathbf{v}^k)\|_2^2 + \lambda_k \|\mathcal{V}(\mathbf{v}^k)\|_1 \tag{5.5}$$

where $\mathbf{z} \in \mathbb{R}^Q$ is the vector of samples of function $u$ and the matrix has a column block structure $\mathbf{\Psi}^k = [\mathbf{\Psi}_{11}^k \ldots \mathbf{\Psi}_{1r_k}^k \mathbf{\Psi}_{21}^k \ldots \mathbf{\Psi}_{2r_k}^k \ldots \mathbf{\Psi}_{r_{k-1}r_k}^k]$, where each column block $\mathbf{\Psi}_{i_{k-1}i_k}^k \in \mathbb{R}^{Q \times n_k}$, $i_{k-1} = 1 \ldots r_{k-1}$, $i_k = 1 \ldots r_k$, is given by

$$(\mathbf{\Psi}_{i_{k-1}i_k}^k)_{qj} = \alpha_{i_{k-1},q}^k \phi_j^k(y_q^k) \beta_{i_k,q}^k, \tag{5.6}$$

for $q = 1, \ldots, Q$ and $j = 1, \ldots, n_k$. Problem (5.5) is solved using the Lasso modified LARS algorithm where the optimal solution is selected using the leave-one-out cross validation procedure presented in algorithm 5. Algorithm 9 outlines the construction of a sparse tensor train approximation using alternating least square.
One of the drawback of alternating least square procedure is that it requires a priori selection of rank vector $r$. Hence finding the optimal rank components may lead to several resolutions of ALS with different guesses of $r$. To overcome this drawback, we introduce in the next section algorithms allowing an automatic selection of the rank.

# 3 Algorithms with automatic rank adaptation

## 3.1 ALS with internal rank adaptation (ALS-RA)

In this method, instead of minimizing $v^k \in \left(\mathcal{S}_{n_k}^k\right)^{r_{k-1} \times r_k}$, we minimize simultaneously $v^k$ and $v^{k+1}$ in rank-$r_k$ canonical subset $\mathcal{R}_{r_k}\left(\left(\mathcal{S}_{n_k}^k\right)^{r_{k-1}} \otimes \left(\mathcal{S}_{n_{k+1}}^{k+1}\right)^{r_{k+1}}\right)$

**Algorithm 9** Algorithm to compute sparse tensor train approximation of a function $u$ using alternating least square.

**Input:** vector of evaluations $\mathbf{z} = (u(y^1), \ldots, u(y^Q))^T \in \mathbb{R}^Q$, rank vector $r$ and tolerance $\epsilon$.

**Output:** sparse tensor train approximation $v = F_{\mathcal{TT}_r^\gamma}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$.

1: Randomly initialize the cores $\mathbf{v}^k$, $k = 1, \ldots, d$. Set $l = 0$ and $v_l = F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$.
2: $l \leftarrow l + 1$.
3: **for** $k = 1, \ldots, d-1$ **do**
4: Compute matrix $\mathbf{\Psi}^k$.
5: Solve (5.5) to obtain $\mathbf{v}^k$.
6: **end for**
7: Set $v_l = F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$
8: **if** $\|v_l - v_{l-1}\| > \epsilon$ and $l \leq l_{max}$ **then**
9: Go to Step 2.
10: **end if**
11: Return $v = v_l$.

with adaptive selection of rank $r_k$. Let us re-write the parametrization of $\mathcal{TT}_r$ as $F_{\mathcal{TT}_r}(\mathbf{v}_{[k,k+1]}, \mathbf{v}^k, \mathbf{v}^{k+1})$. For $k = \{1, \ldots, d-1\}$, we solve

$$\min_{\substack{\mathbf{v}^k \in \mathbb{R}^{r_{k-1} \times r_k \times n_k} \\ \mathbf{v}^{k+1} \in \mathbb{R}^{r_k \times r_{k+1} \times n_{k+1}}}} \|u - F_{\mathcal{TT}_r}(\mathbf{v}_{[k,k+1]}, \mathbf{v}^k, \mathbf{v}^{k+1})\|_Q^2 + \lambda_{r_k}^k \|\mathcal{V}(\mathbf{v}^k)\|_1 + \lambda_{r_k}^{k+1} \|\mathcal{V}(\mathbf{v}^{k+1})\|_1 \tag{5.7}$$

using alternating minimization i.e. we solve alternately the following minimization problems:

$$\min_{\mathbf{v}^k \in \mathbb{R}^{r_{k-1} \times r_k \times n_k}} \|u - F_{\mathcal{TT}_r}(\mathbf{v}_{[k,k+1]}, \mathbf{v}_{r_k}^k, \mathbf{v}_{r_k}^{k+1})\|_Q^2 \quad + \lambda_{r_k}^k \|\mathcal{V}(\mathbf{v}^k)\|_1 \tag{5.8}$$

for fixed $\mathbf{v}^{k+1}$ and

$$\min_{\mathbf{v}^{k+1} \in \mathbb{R}^{r_k \times r_{k+1} \times n_{k+1}}} \|u - F_{\mathcal{TT}_r}(\mathbf{v}_{[k,k+1]}, \mathbf{v}_{r_k}^k, \mathbf{v}_{r_k}^{k+1})\|_Q^2 \quad + \lambda_{r_k}^{k+1} \|\mathcal{V}(\mathbf{v}^{k+1})\|_1 \tag{5.9}$$

for fixed $\mathbf{v}^k$, with selection of models at each step using LARS and fast leave one out cross validation. We solve this minimization problem for several values of $r_k = 1, \ldots, R$ and optimal $r_k$ is selected using $K$-fold cross validation. In the numerical examples, we have used $K = 3$. Algorithm 10 outlines the construction of sparse tensor train approximation using ALS-RA.

## 3.2 Modified alternating least-squares (MALS)

Modified alternating least square consists of performing minimization along two adjacent cores simultaneously and then selecting the rank using truncated SVD.

**Algorithm 10** Algorithm to compute tensor train approximation of a function $u$ using ALS with internal rank adaptation (ALS-RA).

**Input:** vector of evaluations $\mathbf{z} = (u(y^1), \ldots, u(y^Q))^T \in \mathbb{R}^Q$, initial rank vector $r \in \mathbb{N}^{d+1}$ with $r_0 = 1 = r_d$, maximum rank $R$ and tolerance $\epsilon$.
**Output:** sparse tensor train approximation $v = F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$.

1: Randomly initialize the cores $\mathbf{v}^k$, $k = 1, \ldots, d$. Set $l = 0$ and $v_l = F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$.
2: $l \leftarrow l + 1$.
3: **for** $k = 1, \ldots, d-1$ **do**
4:  **for** $r_k = 1, \ldots, R$ **do**
5:  Solve (5.7) using alternating least squares algorithm (with model selection in each iteration).
6:  **end for**
7:  Select optimal $r_k \in \{1, \ldots, R\}$ using cross validation.
8: **end for**
9: Set $v_l = F_{\mathcal{TT}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$
10: **if** $\|v_l - v_{l-1}\| > \epsilon$ and $l \leq l_{max}$ **then**
11:  Go to Step 2.
12: **end if**
13: Return $v = v_l$.

For $k \in \{1, \ldots, d-1\}$, let $r_{[k]} = (r_j)_{j \neq k}$, the subset $\mathcal{TT}_{r_{[k]}}$ is parametrized by the map $F_{\mathcal{TT}_{r_{[k]}}} : \mathcal{P} \to \mathcal{S}_n$, $\mathcal{P} = \mathcal{P}^1 \times \ldots \times \mathcal{P}^{k-1} \times \mathcal{W}^{k,k+1} \times \mathcal{P}^{k+2} \ldots \times \mathcal{P}^d$, $\mathcal{P}^j = \mathbb{R}^{r_{j-1} \times r_j \times n_j}$, $j \in \{1, \ldots, d-1\} \setminus \{k, k+1\}$ and $\mathcal{W}^{k,k+1} = \mathbb{R}^{r_{k-1} \times n_k \times r_k \times n_{k+1}}$ such that $v \in \mathcal{TT}_{r_{[k]}}$ is written

$$v = F_{\mathcal{TT}_{r_{[k]}}}(\mathbf{v}^1, \ldots \mathbf{v}^{k-1}, \mathbf{w}^{k,k+1}, \mathbf{v}^{k+2}, \ldots, \mathbf{v}^d),$$

where the *supercore* $\mathbf{w}^{k,k+1}$ correspond to the coefficient of bivariate functions given by $w^{k,k+1}_{i_{k-1}i_{k+1}} \in \mathcal{S}^k_{n_k} \otimes \mathcal{S}^{k+1}_{n_{k+1}}$. The $r_{k-1} \times r_{k+1}$ bivariate function $w^{k,k+1}_{i_{k-1}i_{k+1}}$ can be represented as $w^{k,k+1}_{i_{k-1}i_{k+1}} = \langle \mathbf{w}^{k,k+1}_{i_{k-1}i_{k+1}}, \boldsymbol{\phi}^{k,k+1} \rangle$, where $\mathbf{w}^{k,k+1}_{i_{k-1}i_{k+1}}$ is the set of coefficients in the basis $\{\phi^k_i \otimes \phi^{k+1}_j : 1 \leq i \leq n_k,\ 1 \leq j \leq n_{k+1}\}$ of $\mathcal{S}^k_{n_k} \otimes \mathcal{S}^{k+1}_{n_{k+1}}$. The supercore $\mathbf{w}^{k,k+1}$ thus parametrizes $w^{k,k+1} \in \left(\mathcal{S}^k_{n_k} \otimes \mathcal{S}^{k+1}_{n_{k+1}}\right)^{r_{k-1} \times r_{k+1}}$. For simplicity of notation, we will denote $F_{\mathcal{TT}_{r_{[k]}}}(\mathbf{v}^1, \ldots \mathbf{v}^{k-1}, \mathbf{w}^{k,k+1}, \mathbf{v}^{k+2}, \ldots, \mathbf{v}^d)$ by $F_{\mathcal{TT}_{r_{[k]}}}(\mathbf{v}_{[k,k+1]}, \mathbf{w}^{k,k+1})$. In MALS algorithm, for each $k \in \{1, \ldots, d-1\}$, we perform the following two steps.

**Minimization Step:** For each $k \in \{1, \ldots, d-1\}$, we solve the minimization problem given by

$$\min_{\mathbf{w}^{k,k+1} \in \mathcal{W}^{k,k+1}} \|u - F_{\mathcal{TT}_{r_{[k]}}}(\mathbf{v}_{[k,k+1]}, \mathbf{w}^{k,k+1})\|^2_Q + \lambda_k \|\mathcal{V}(\mathbf{w}^{k,k+1})\|_1,$$

where $\mathcal{V}(\mathbf{w}^{k,k+1}) \in \mathbb{R}^{(r_{k-1}n_k r_{k+1}n_{k+1})}$ is a vectorization of $\mathbf{w}^{k,k+1}$. This gives the least squares problem

$$\min_{\substack{\mathbf{w}^{k,k+1}\in\mathcal{W}^{k,k+1}\\ \|\mathcal{V}(\mathbf{w}^{k,k+1})\|_1\leq\gamma_k}} \frac{1}{Q}\sum_{q=1}^{Q}\left(u(y_q) - \left\langle \mathbf{w}^{k,k+1}, \alpha_q^k \otimes \phi^k(y_q^k)\otimes\beta_q^{k+1}\otimes\phi^{k+1}(y_q^{k+1})\right\rangle\right)^2. \quad (5.10)$$

Problem (5.10) can be equivalently written in the form

$$\min_{\hat{\mathbf{w}}^{k,k+1}\in\mathbb{R}^{r_{k-1}\times n_k\times r_k\times n_{k+1}}} \|\mathbf{z} - \mathbf{H}^{k,k+1}\mathcal{V}\left(\hat{\mathbf{w}}^{k,k+1}\right)\|_Q^2 + \lambda_k\|\mathcal{V}\left(\hat{\mathbf{w}}^{k,k+1}\right)\|_1. \quad (5.11)$$

Here the matrix has a column block structure $\mathbf{H}^{k,k+1} = [\mathbf{H}_{11}^{k,k+1}\dots\mathbf{H}_{1r_{k+1}}^{k,k+1}\mathbf{H}_{21}^{k,k+1}\dots\mathbf{H}_{2r_{k+1}}^{k,k+1}\dots\mathbf{H}_{r_{k-1}r_{k+1}}^{k,k+1}]$, where each column block $\mathbf{H}_{i_{k-1}i_{k+1}}^{k,k+1} \in \mathbb{R}^{Q\times(n_k n_{k+1})}$, $i_{k-1}\in\{1\dots r_{k-1}\}$, $i_{k+1}\in\{1\dots r_{k+1}\}$, is given by

$$(\mathbf{H}_{i_{k-1}i_{k+1}}^{k,k+1})_{qj} = \alpha_{i_{k-1},q}^k \phi_j^{k,k+1}(y_q^k, y_q^{k+1})\beta_{i_{k+1},q}^{k+1}. \quad (5.12)$$

Problem (5.11) can be solved using lasso modified LARS with selection of $\lambda_k$ using fast leave one out cross validation. We can thus obtain $\mathbf{w}^{k,k+1}$.

**Low rank truncation:** In this step we wish to recover the core tensors $\mathbf{v}^k$ and $\mathbf{v}^{k+1}$ from $\mathbf{w}^{k,k+1}$. For this purpose, we reshape $\mathbf{w}^{k,k+1}$ as a matrix $\mathbf{W}^{k,k+1} \in \mathbb{R}^{(r_{k-1}n_k)\times(r_{k+1}n_{k+1})}$ such that $\mathbf{W}^{k,k+1}$ parametrizes $W^{k,k+1} \in \left(\mathcal{S}_{n_k}^k\right)^{r_{k-1}} \otimes \left(\mathcal{S}_{n_{k+1}}^{k+1}\right)^{r_{k+1}}$. We then by perform a truncated singular value decomposition of this matrix:

$$\mathbf{W}^{k,k+1} \approx \sum_{i_k=1}^{r_k} \sigma_{i_k}\mathbf{V}_{i_k}^k \otimes \mathbf{V}_{i_k}^{k+1},$$

with adaptive selection of rank $r_k$. $\{\mathbf{V}_{i_k}^k\}^{r_k}$ provides coefficient vectors of $r_{k-1}\times r_k$ functions $v_{i_{k-1}i_k}^k \in \mathcal{S}_{n_k}^k$ and $r_k \times r_{k+1}$ functions $v_{i_k i_{k+1}}^{k+1} \in \mathcal{S}_{n_{k+1}}^{k+1}$. The vectors $\{\sigma_{i_k}\mathbf{V}_{i_k}^k\}_{i_k=1}^{r_k}, \mathbf{V}_{i_k}^k \in \mathbb{R}^{r_{k-1}n_k}$ can be reshaped as tensor train core $\mathbf{v}^k \in \mathbb{R}^{r_{k-1}\times r_k\times n_k}$. Similarly $\{\mathbf{V}_{i_k}^{k+1}\}_{i_k=1}^{r_k} \in (\mathbb{R}^{r_{k+1}n_{k+1}})^{r_k}$ can be reshaped as $\mathbf{v}^{k+1} \in \mathbb{R}^{r_k\times r_{k+1}\times n_{k+1}}$. Algorithm 11 outlines the construction of a sparse tensor train approximation using MALS.

The selection of $r_k$ might be based on

$$\|\mathbf{W}^{k,k+1} - \sum_{i_k=1}^{r_k}\sigma_{i_k}\mathbf{V}_{i_k}^k \otimes \mathbf{V}_{i_k}^{k+1}\|_F \leq \varepsilon \quad (5.13)$$

for some small $\varepsilon > 0$. In general, the optimal value of $\varepsilon$ may not be known *a priori*. Moreover, for a given sample size, we may not be able to achieve a given value of

**Algorithm 11** Algorithm to compute sparse tensor train approximation of a function $u$ using MALS.

**Input:** vector of evaluations $\mathbf{z} = (u(y^1), \ldots, u(y^Q))^T \in \mathbb{R}^Q$, rank vector $r \in \mathbb{N}^{d+1}$ and tolerance $\epsilon \in \mathbb{R}_+$.

**Output:** sparse tensor train approximation $v = F_{\mathcal{T}\mathcal{T}_r^\gamma}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$.

1: Randomly initialize the cores $\mathbf{v}^k$, $k = 1, \ldots, d$. Set $l = 0$ and $v_l = F_{\mathcal{T}\mathcal{T}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$.
2: $l \leftarrow l + 1$.
3: **for** $k = 1, \ldots, d-1$ **do**
4: Compute matrix $\mathbf{H}^{k,k+1}$.
5: Solve problem (5.11) for input $\mathbf{z} \in \mathbb{R}^Q$ to obtain $\mathbf{w}^{k,k+1}$.
6: Reshape vector $\mathbf{w}^{k,k+1}$ as matrix $\mathbf{W}^{k,k+1}$. Compute truncated SVD with selection of rank $r_k$
7: Reshape $\{\sigma_{i_k}\mathbf{V}_{i_k}^k\}_{i_k=1}^{r_k}$ to get $\mathbf{v}^k$ and reshape $\{\mathbf{V}_{i_k}^{k+1}\}_{i_k=1}^{r_k}$ to get $\mathbf{v}^{k+1}$.
8: **end for**
9: $v_l = F_{\mathcal{T}\mathcal{T}_r}(\mathbf{v}^1, \ldots, \mathbf{v}^d)$.
10: **if** $\|v_l - v_{l-1}\| > \epsilon$ and $l \leq l_{max}$ **then**
11: Go to Step 2.
12: **end if**
13: Return $v = v_l$.

$\varepsilon$. Hence, instead of choosing $\varepsilon$, we use $K$-fold cross validation to choose $r_k$. In the numerical examples in section 4, the optimal value of $r_k$ is chosen by 3-fold cross validation.

Classical MALS may suffer from certain disadvantages. The total number of coefficients to be determined in solving (5.11) is given by $r_{k-1} \times n_k \times r_k \times n_{k+1}$. If few sample evaluations are available such that $Q \ll r_{k-1} \times n_k \times r_k \times n_{k+1}$, we may not be able to determine optimal sparse coefficient vector using least square with sparse regularization. However, as will be shown in the illustrations, MALS gives better accuracy if sufficiently enough samples are available.

# 4 Illustrations

## 4.1 Analytical model: Ishigami function

Let us first consider the so called Ishigami function which is widely used for benchmarking in global sensitivity analysis [9]:

$$u(\xi) = \sin(\xi_1) + 7\sin^2(\xi_2) + 0.1\xi_3\sin(\xi_1),$$

where $\xi_i (i = 1, \ldots, 3)$ are uniformly distributed random variables over $[-\pi, \pi]$. We use an orthonormal basis composed of Legendre polynomials. The Ishigami function has sparse representation on this basis.

### 4.1.1 Results using Alternating Least-Squares

We will first consider the performance of ALS (algorithm 9). As noted in section 2, one of the drawback of ALS algorithm is the selection of optimal rank $r$. Since the optimal $r$ is not known *a priori*, one needs to perform ALS for several guesses of tensor train rank. This is particularly significant for high-dimensional problems where the combinatorial problem of finding the optimal rank can be computationally expensive and may not be feasible. Table 5.1 shows the approximation errors obtained with several rank vectors using $Q = 300$ samples in the approximation space $\otimes_{k=1}^{3}\mathbb{P}_{10}^{k}$. We can compare the error $\varepsilon(u_{tt}, u)$ obtained with optimal $r$, which in this case is [ 1 3 6 1 ], with other choices of $r$. We find that the choice of an optimal rank vector $r$ (which also depends on the sample size and sample set) is critical in obtaining an accurate approximation.

**Table 5.1:** Ishigami function: Relative error $\varepsilon(u_{tt}, u)$ w.r.t different choice of rank vector $r$ using Algorithm 9 with $Q = 300$ samples in the approximation space $\otimes_{k=1}^{3}\mathbb{P}_{10}^{k}$.

| TT rank ($r$) | Error ($\varepsilon(u_{tt}, u)$) |
|---|---|
| [ 1 1 1 1 ] | 1.0051 |
| [ 1 2 2 1 ] | $4.48 \times 10^{-1}$ |
| [ 1 6 6 1 ] | $6.03 \times 10^{-1}$ |
| [ 1 8 8 1 ] | $1.1 \times 10^{-3}$ |
| [ 1 5 7 1 ] | $9.2 \times 10^{-4}$ |
| [ 1 3 6 1 ] | $8.4 \times 10^{-4}$ |

### 4.1.2 Results using Modified Alternating Least-Squares

We illustrate the results obtained with Algorithm 11 in Table 5.2 for different sample size $Q$ and polynomial degree $p$ together with the optimal $r$ selected by the algorithm. The approximation error $\varepsilon(u_{tt}, u)$ reduces with increase in sample size $Q$. Note that the algorithm automatically selects the rank and gives an approximation with higher accuracy than with ALS.

## 4.2 Analytical model: Sine of a sum

The purpose of this example is to illustrate that for certain functions, approximations obtained in sparse TT format outperforms canonical format. We consider the function

$$u(\xi) = \sin(\xi_1 + \xi_2 + \ldots + \xi_6) \tag{5.14}$$

**Table 5.2:** Ishigami function: Relative error $\varepsilon(u_{tt}, u)$ and optimal rank $r$ obtained by Algorithm 6

| | Q=100 | | Q=200 | |
|---|---|---|---|---|
| $p$ | $\varepsilon(u_{tt}, u)$ | TT rank $(r)$ | $\varepsilon(u_{tt}, u)$ | TT Rank $(r)$ |
| 8 | 0.0304 | [ 1 3 5 1 ] | 0.01 | [ 1 8 3 1 ] |
| 10 | 0.0023 | [ 1 2 3 1 ] | $9.01 \times 10^{-4}$ | [ 1 2 2 1 ] |
| 12 | 0.2415 | [ 1 12 13 1 ] | $5.61 \times 10^{-5}$ | [ 1 2 2 1 ] |
| 14 | 0.2543 | [ 1 13 13 1 ] | $4.66 \times 10^{-6}$ | [ 1 2 2 1 ] |

where $\xi$ are uniform random variables over $[-1, 1]$. This function is also considered in [58] and is shown to have a low rank ($r_k = 2$ for $1 \leq k \leq d-1$) tensor train representation and thus is a good example to illustrate the applicability of this format. In figure 5.1, we compare the solution obtained using Algorithm 10, classical least square method using sparse regularization, greedy construction in $\mathcal{R}_m$ and direct approximation in $\mathcal{R}_m$ (see chapter 4) using different sample size $Q$ in the approximation space $\otimes_{k=1}^{6} \mathbb{P}_p^k$ with selection of optimal $p$ using cross validation. We find that, for this function, the approximation obtained by algorithm 10 with few samples gives very accurate approximation as compared to other algorithms using canonical format. In figure 5.2, we plot the evolution of the ranks v/s ALS iterations

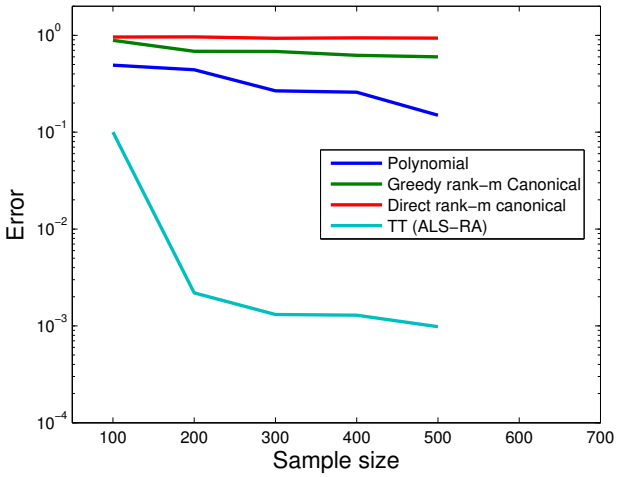



**Figure 5.1:** Sine of a sum: Evolution of error v/s sample size using algorithm 10 using optimal polynomial degree $p$ in the approximation space $\otimes_{k=1}^{6} \mathcal{S}_p^k$

using $Q = 300$ samples and polynomial degree $p = 3$. We find that with algorithm 10, after a few iterations, ALS converges to TT approximation with stabilized $r$. In this benchmark example, we illustrated the following:

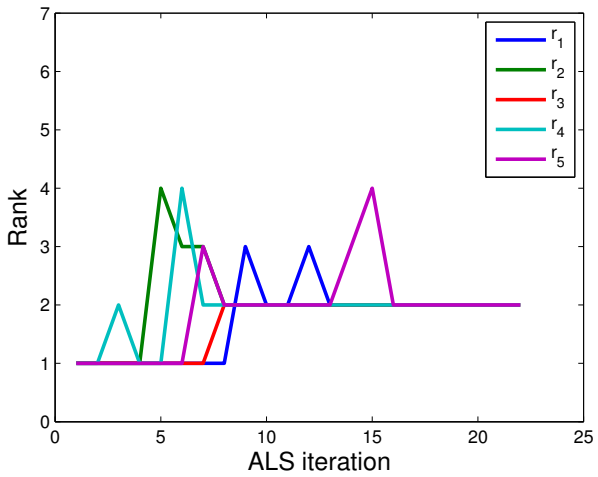



**Figure 5.2:** Sine of a sum: Evolution of rank components $r_i, 1 \leq i \leq 5$, v/s ALS-RA iterations using algorithm 11 using $Q = 300$ and $p = 3$

- For functions having a low rank TT representation, approximation in sparse TT format with ALS-RA gives a more accurate representation as compared to canonical sparse tensor formats for the same sample size and approximation space.

- Given enough samples, ALS-RA is able to recover TT rank of the function.

## 4.3 Application example: Borehole function

Let us consider the borehole function also considered in [81]:

$$f(\xi) = \frac{2\pi\xi_3(\xi_4 - \xi_6)}{\ln(\xi_2/\xi_1)\left(1 + \frac{2\xi_7\xi_3}{\ln(\xi_2/\xi_1)\xi_1^2\xi_8} + \frac{\xi_3}{\xi_5}\right)}.$$

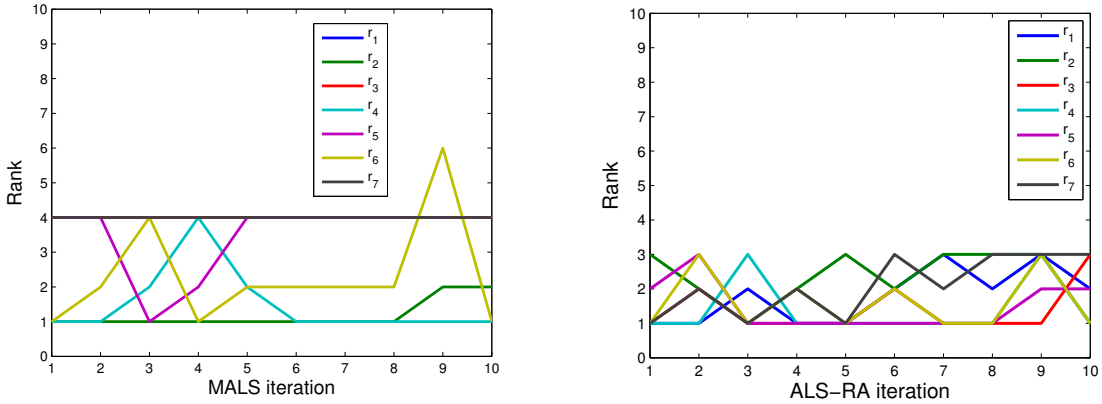
This function takes 8 parametric inputs and their distributions are indicated in Table 5.3.

In this high-dimensional example, we compare the performance of MALS (algorithm 11) and ALS-RA (algorithm 10). In fig 5.3 we plot the evolution of TT rank components with iterations for sample size $Q = 200$ in the approximation space $\otimes_{k=1}^{8}\mathbb{P}_p$ with optimal selection of $p \in \{2, \ldots, 4\}$ using cross validation. We find that both MALS and ALS-RA algorithms select small TT rank components. In fig. 5.4(a), we plot the sparsity ratio of the *supercores* $\mathbf{w}^{k,k+1}, 1 \leq k \leq 7$, w.r.t MALS iterations. Similarly, in fig. 5.4(b), we plot the sparsity ratio of the *cores* $\mathbf{v}^k, 1 \leq k \leq 8$ w.r.t ALS-RA iterations. The MALS algorithm gives better sparsity ratios as compared to classical ALS-RA. Since MALS involves computing a sparse representation of the supercore $\mathbf{w}^{k,k+1}$ (i.e. coefficients corresponding to bivariate basis $\boldsymbol{\phi}^k \otimes \boldsymbol{\phi}^{k+1}$), it is

**Table 5.3:** Random variable inputs to Borehole function and their corresponding distributions

| RV | Description | Distribution |
|---|---|---|
| $\xi_1$ | radius of borehole (m) | $N(\mu = 0.10, \sigma = 0.0161812)$ |
| $\xi_2$ | radius of influence (m) | $LN(\mu = 7.71, \sigma = 1.0056)$ |
| $\xi_3$ | transmissivity of upper aquifer ($m^2$/yr) | $U(63070, 115600)$ |
| $\xi_4$ | potentiometric head of upper aquifer (m) | $U(990, 1110)$ |
| $\xi_5$ | transmissivity of lower aquifer ($m^2$/yr) | $U(63.1, 116)$ |
| $\xi_6$ | potentiometric head of lower aquifer (m) | $U(700, 820)$ |
| $\xi_7$ | length of borehole (m) | $U(1120, 1680)$ |
| $\xi_8$ | hydraulic conductivity of borehole (m/yr) | $U(9855, 12045)$ |

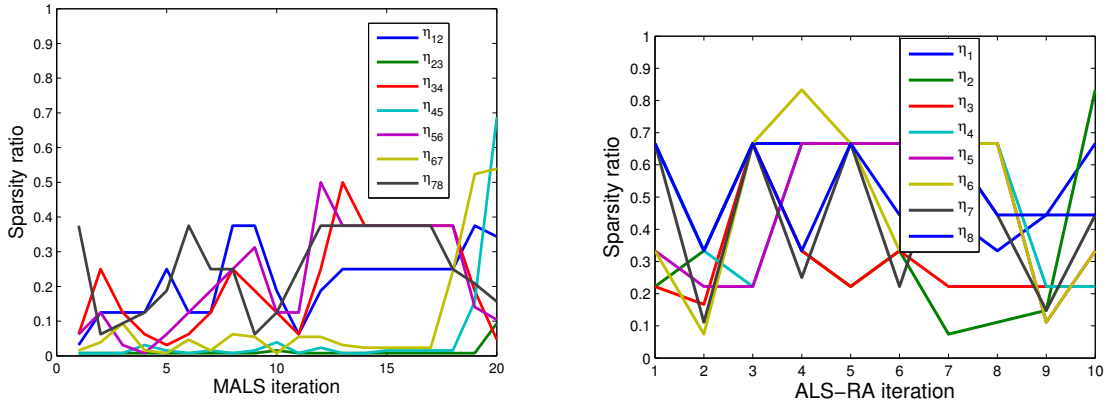
better able to better exploit the sparsity of the function than ALS-RA in which we still compute coefficients corresponding to univariate basis $\phi^k$ and $\phi^{k+1}$ alternately.




(a) (b)

**Figure 5.3:** Borehole function: Evolution of tensor train ranks v/s iterations using (a) MALS and (b) ALS-RA for $Q = 200$. Note that sparsity ratios plotted in (a) are of *supercores* $\mathbf{w}^{k,k+1}$ and in (b) are of *cores* $\mathbf{v}^k$ and hence not comparable.

We compare MALS and ALS-RA for different sample size in fig 5.5. We consider both these algorithms without regularization (ordinary least square (OLS)) and with sparse regularization. We find that, with OLS, both the algorithms give inaccurate approximations for few samples ($Q < 300$). For $Q \geq 300$, ALS-RA gives better accuracy. Indeed, given enough samples, ALS-RA better estimates (comparatively) few coefficients in each iteration as compared to MALS. However, with regularization, MALS gives better accuracy as compared ALS-RA since it better exploits the sparsity of the borehole function as shown in 5.4.




(a) (b)

**Figure 5.4:** Borehole function: Evolution of sparsity ratio v/s iterations using (a) MALS and (b) ALS-RA for $Q = 200$

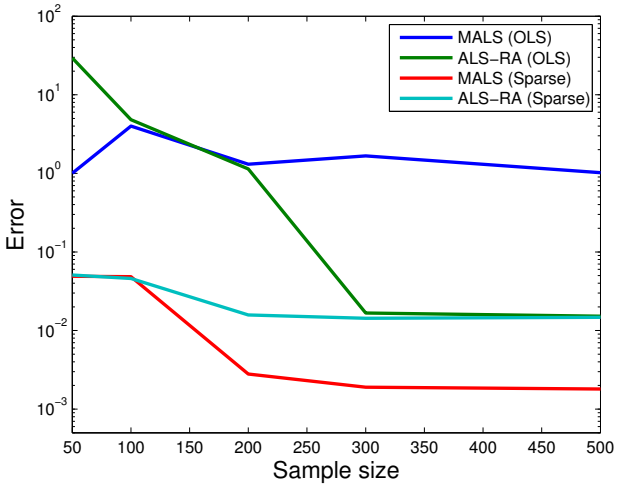



**Figure 5.5:** Borehole function: Comparison of MALS and ALS-RA

## 4.4 Application example: A stochastic partial differential equation (Canister)

The final example (also considered in [62]) represents the transport of pollutant inside an active carbon filter. In this example, the transient convection-reaction-diffusion equation ((5.15)) is solved in the simplified geometry shown in Figure 5.6(a).

$$\frac{\partial u}{\partial t} - \nabla(\kappa \nabla u) + c(D \cdot \nabla u) = \sigma u. \tag{5.15}$$

The random parameters in this model and their distribution are shown in Table 5.4. Boundary conditions are Neumann homogeneous due to impermeability of the

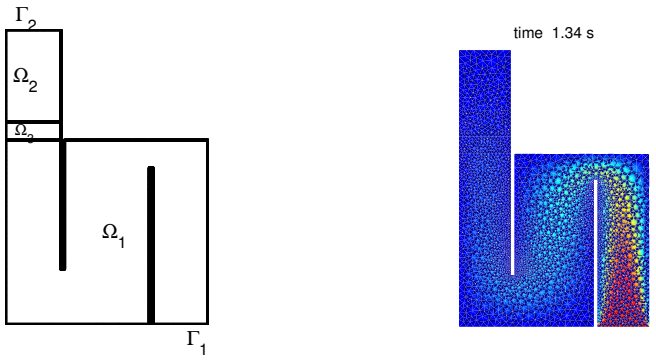
**Table 5.4:** Random variable inputs to Canister problem and their corresponding distributions

| RV | Parameter | Distribution |
|---|---|---|
| $\xi_1$ | $u(t=0)$ | $U(0.8, 1.2)$ on $\Omega$ |
| $\xi_2$ | $\sigma$ | $U(8, 12)$ on $\Omega_2$ |
| $\xi_3$ | $\sigma$ | $U(0.8, 1)$ on $\Omega_1$ |
| $\xi_4$ | $c$ | $U(1, 5)$ |
| $\xi_5$ | $\kappa$ | $U(0.02, 0.03)$ |

canister wall and are given by

$$u = \xi_1 \text{ on } \Gamma_1 \times \Omega_t, \tag{5.16}$$
$$u = 0 \text{ on } \Gamma_2 \times \Omega_t. \tag{5.17}$$

The concentration of pollutants $u$ for one realization of samples is shown in Figure 5.6(b)




(a) Domain (b) Finite element solution

**Figure 5.6:** Canister: (a)Computational domain and (b)representive finite element solution for one realization of random inputs

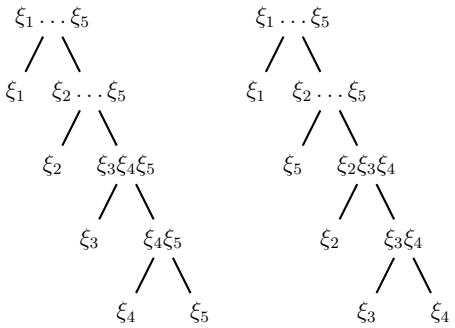
The quantity of interest is the total amount of pollutant captured by the filter domain $\Omega_3$ over the time interval. This quantity is expressed as

$$I(u) = \int_t \int_{\Omega_3} u(x,t) dx dt.$$

In this numerical example, we compare the performance of TT tensor formats associated with two different trees shown in figure 5.7. Indeed, TT format is a particular

case of Hierarchical Tensor format and depending on ordering of variables, we can obtain different approximations for the same sample size and sample set. However, the selection of a suitable tree for optimal TT representation is problem dependant and is difficult to determine *a priori*. Adaptive strategies for selection of optimal trees for a given sample set is a subject of future research. As in example 4.2, in figure 5.8, we compare the performance of different algorithms for different sample sizes in approximation space $\otimes_{k=1}^{5}\mathbb{P}_p$, where optimal $p \in \{1, \ldots, 4\}$ is selected using cross validation. In this example, we observe the following:

- We obtain two different approximations corresponding to Tree 1 and Tree 2. This confirms that approximations have a dependence on ordering of variables.
- In this example, approximation in TT format gives better approximation as compared to canonical tensor format (both greedy and direct rank-m approximations).




(a) Tree 1 (b) Tree 2

**Figure 5.7:** Canister: Two different trees associated with TT representation

# 5 Conclusion

In this chapter, we detailed the implementation of sparse tensor train approximation. Algorithms based on ALS and MALS for approximation in sparse low rank tensor subsets were presented and their performance was illustrated on analytical and numerical examples. Particularly, it was illustrated that inducing the sparsity in TT representation leads to selection of smaller TT ranks with small components thus leading to better approximation with available information. Also, the TT format is more promising compared to canonical tensor format as illustrated in numerical examples. Note that the TT format is based on separated representation and hence calculation of statistical quantities like mean, standard deviation and Sobol sensitivity indices from TT format is straightforward. One of the future work is to select an optimal tree for TT approximation with available sample based information. Also, more general tree based formats could be considered.

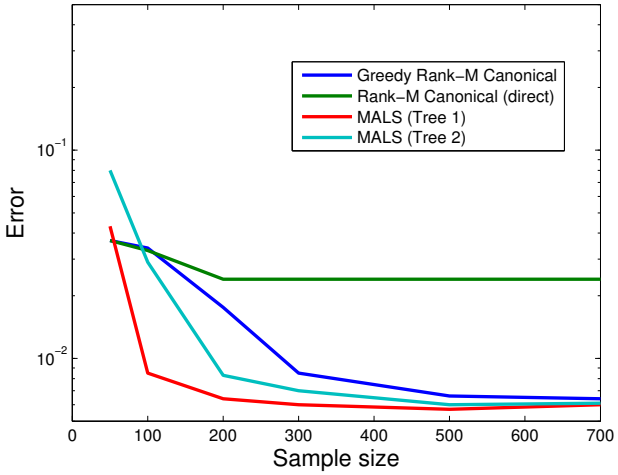



**Figure 5.8:** Canister: Evolution of error v/s sample size using different algorithms with optimal polynomial degree $p$ in the approximation space $\otimes_{k=1}^{6}\mathbb{P}_p$

# Chapter 6

# Clustering and Classification based on Low Rank Approximation

*In this chapter, we combine ideas of clustering and classification with structured approximation (sparse polynomial or low rank approximation) to be able to approximate discontinuous and irregular functions. The method relies on a clustering scheme in which the sample points are sorted based on geometric distance in a set of clusters. We then perform a re-partition of points within clusters based on an iterative scheme that minimizes residuals in each cluster. Finally we perform merging of clusters based on an error criterion. A new sample point is distributed in one of the final clusters based on a classification algorithm.*
*Initial approach has been suggested by O. Le Maitre and O. Knio. Here, we bring two additional contributions. The first contribution is the use of fast leave one out cross validation error for the estimation of residuals and the second is the use of sparse polynomial and low rank approximation methods instead of classical least squares based polynomial approximation.*

## Contents

# 1 Introduction

Our aim is to approximate a function $u : \xi \in \Xi \mapsto u(\xi) \in \mathbb{R}$. We wish to devise a strategy that can identify $L$ sub-domains $\Xi_l$ such that $\Xi = \cup_l \Xi_l$, $\Xi_l \cap \Xi_{l'} = \emptyset$ for $l \neq l'$ over which we can obtain good approximations of $u$. The function $u$ can then be approximated locally by means of approximations $\hat{v}_l, 1 \leq l \leq L$ such that

$$u(y) \approx \sum_{l=1}^{L} \hat{v}_l(y) 1_{\Xi_l}(y), \tag{6.1}$$

where $1_{\Xi_l}$ is the indicator function of the set $\Xi_l$:

$$1_{\Xi_l}(y) = \begin{cases} 1, \ y \in \Xi_l \\ 0, \text{otherwise.} \end{cases} \tag{6.2}$$

If $u$ is non-smooth across the sub-domains boundary, the approximation error of (6.1) is expected to be smaller than direct approximation over $\Xi$ using standard approximations (e.g. polynomial).

# 2 Clustering Method

To construct piecewise approximation of $u$, we consider a set of $Q$ samples consisting of distinct couples $\{y^q, u(y^q)\}$, where $\mathcal{Q} = \{1, \ldots, Q\}$ is the set of observation indices.

## 2.1 Cluster Initialization

Since the actual minimal number $L$ of sub-domains over which $u$ can be well approximated is not known a priori, the first step is to partition the sample set into subsets or clusters. We denote by $K$ the initial number of clusters. Each cluster contains a subset of samples, $\mathcal{K}_k \subset \mathcal{Q}$, while a sample belongs to a unique cluster. That is

$$\cup_{k=1}^{K} \mathcal{K}_k = \mathcal{Q} \text{ and } \mathcal{K}_k \cap \mathcal{K}_{k'} = \emptyset \text{ for } k \neq k'. \tag{6.3}$$

We will denote $k_q$ the membership of $y^q, q \in \mathcal{Q} : k_i = k \Leftrightarrow q \in \mathcal{K}_k$. The initial clustering of the observations is based on minimizing the overall geometric (Euclidean) intra-cluster distance $G$, defined by

$$G^2 = \sum_{k=1}^{K} G^2(\mathcal{K}_k), \ G^2(\mathcal{K}_k) = \sum_{q \in \mathcal{K}_k} \|y^q - Y_k\|^2, \ Y_k = \frac{1}{\#\mathcal{K}_k} \sum_{q \in \mathcal{K}_k} y^q. \tag{6.4}$$

Here $Y_k$ is the barycentric center of the cluster $k$. The exact minimization of $G^2$ is a combinatorial problem whose solution can be approximated by a classical iterative algorithm consisting of sequence of updates in the sample memberships, affecting each of them to the cluster having the closest centroid $Y_k$, followed by updates of cluster centroids. Algorithm 12 outlines the geometric clustering scheme.

**Algorithm 12** Geometric Clustering

**Input:** $K, \{y^q, q \in \mathcal{Q}\}$. {Number of Clusters and samples}
**Output:** Clusters $\mathcal{K}_k, 1 \leq k \leq K$.

**for** $k = 1, \ldots, K$ **do**
  Set at random $Y_k \in \Xi$ {Random initialization of centroids}
**end for**
**while** Not converged **do**
  **for** $k = 1, \ldots, K$ **do**
    $\mathcal{K}_k \leftarrow \emptyset$ {Initialization of clusters}
  **end for**
  **for** $q = 1, \ldots, Q$ **do**
    $k \leftarrow \text{argmin}_l \|y^q - Y_l\|^2$ {Find closest centroid}
    $\mathcal{K}_k \leftarrow \mathcal{K}_k \cup \{q\}$ {Affect observation to its clusters}
  **end for**
  **for** $k = 1, \ldots, K$ **do**
    $Y_k \leftarrow \frac{1}{\#\mathcal{K}_k} \sum_{q \in \mathcal{K}_k} y^q$ {Recompute Centroids}
  **end for**
**end while**

## 2.2 Approximation models

For the sample set and its $K-$clustering, we construct approximation model in each cluster. In this work, for each cluster $k$, we have considered two choices for constructing cluster models $\hat{v}_k$:

- Sparse polynomial model
- Sparse low rank model

In section 3, we will illustrate both these approaches. Let us define

$$R^2 = \sum_{k=1}^{K} R^2(\mathcal{K}_k), \; R^2(\mathcal{K}_k) = \sum_{q \in \mathcal{K}_k} |u(y^q) - \hat{v}_k(y^q)|^2, \tag{6.5}$$

as the total (squared) residual error associated with the $K-$ clustering of the samples.

## 2.3 Clustering scheme

We now wish to update the clusters $\{\mathcal{K}_k\}$ such that in each cluster, we can obtain sufficiently accurate approximation model. The geometric clustering algorithm 12 aims at minimizing the inter-cluster distance, it is likely that it results in some clusters overlapping with irregularities in $u$, with consequently significant residual error. However, we seek clusters that minimize the sum of squared residuals, but

at the same time, are distinct and do not overlap with each other. Thus to prevent cluster overlapping, we introduce a penalty term accounting for the inter-cluster distance $G^2$. The functional considered for minimization is therefore

$$\hat{R}^2 = R^2 + \gamma G^2 = \sum_{k=1}^{K} \left( \sum_{q \in \mathcal{K}_k} |u(y^q) - \hat{v}_k(y^q)|^2 + \gamma \|y^q - Y_k\|^2 \right), \tag{6.6}$$

for some $\gamma > 0$. The largest $\gamma$ the tightest is the constraint on the cluster geometry, with lower chance of obtaining clusters belonging to single sub-domain $\Xi_l$ over which we can obtain accurate approximation. A smaller value of $\gamma$ allows clusters to adapt the features of $u$, but $\gamma \to 0$ can lead to mixing of clusters. Re-writing (6.6) using sample memberships, we have

$$\hat{R}^2 = \sum_{q \in \mathcal{Q}} \left( |u(y^q) - \hat{v}_{k_q}(y^q)|^2 + \gamma \|y^q - Y_{k_q}\|^2 \right), \tag{6.7}$$

and it will be convenient to introduce the squared residual $r^2_{q,k}$ as a short-hand notation for

$$r^2_{q,k} = |u(y^q) - \hat{v}_k(y^q)|^2 + \gamma \|y^q - Y_k\|^2. \tag{6.8}$$

Based on (6.6), a basic iterative scheme would be to sequentially a) update the observation memberships and b) update the cluster models using the updated memberships. For the update of the memberships, we consider the following rule

$$\text{if } \min_{k \in \{1,\dots,K\}} r^2_{q,k} < r^2_{q,k_q}, \text{ then } k_q \leftarrow \arg\min_{k \in \{1,\dots,K\}} r^2_{q,k}. \tag{6.9}$$

That is, the membership of an observation is updated to a new cluster only if it has a (strictly) lower squared residual.

It should be noted that, due to over-fitting $r_{q,k}$ will in general have lowest value for cluster $k_q$. The way to circumvent this problem is to use fast leave one out error estimate for each $q$ in $\mathcal{K}_k$. For a given $q \in \mathcal{Q}$, if $k \neq k_q$, we use the formula (6.8) (the estimated model did not use sample $q$) but for $k = k_q$, we replace (6.8) by

$$r^2_{q,k} = |u(y^q) - \hat{v}_k^{\setminus q}(y^q)|^2 + \gamma \|y^q - Y_k\|^2, \tag{6.10}$$

where $\hat{v}_k^{\setminus q}(y^q)$ is the model estimated using samples $\mathcal{K}_k \setminus \{q\}$. In practice, $|u(y^q) - \hat{v}_k^{\setminus q}(y^q)|$ is computed using fast leave one out (see chapter 3, section 4). However, due to oscillatory nature of the polynomial basis, we can obtain low squared residual values, $|u(y^q) - \hat{v}_k(y^q)|$, at observation points that are not close to the cluster $k$. Hence, direct application of the rule (6.9) may lead to disconnected clusters with isolated members. This can be avoided by constraining the membership updates to some neighbourhood of the clusters, such that their shapes evolve progressively through changes at their boundaries. Thus, we can store the indices of $m-$nearest

sample points to $y^q$ and check that at least one of this point belong to $\mathcal{K}_k$ to accept an update to $k$ of the membership. The complete rule for membership update is finally:

$$\text{if } \min_{k\in\mathcal{N}_m(y^q)} r_{q,k}^2 < r_{q,k_q}^2, \text{ then } k_q \leftarrow \arg\min_{k\in\mathcal{N}_m(y^q)} r_{q,k}^2, \tag{6.11}$$

where $\mathcal{N}_m(y^q)$ is the set of cluster memberships of $m-$neighbours of $y^q$. The resulting model clustering iterative scheme is outlined in Algorithm 13.

**Algorithm 13** Model-Based clustering algorithm

**Input:** $\{(y^q, u(y^q)), q \in \mathcal{Q}\}$, $\{\mathcal{K}_k, k = 1, \ldots, K\}$, $\gamma > 0$ and $m > 0$ {Observations and initial clusters sets}

**for** $k = 1, \ldots, K$ **do**
  Compute approximation $\hat{v}_k$ using samples $\mathcal{K}_k$ {Set cluster models}
  **for all** $q \in \mathcal{K}_k$ **do**
    $k_q \leftarrow k$ {Set observation memberships}
  **end for**
**end for**
**while** Not converged **do**
  **for all** $q \in \mathcal{Q}$ **do**
    Find $k \in \mathcal{N}_m(y^q)$, minimizing $r_{q,k}^2$
    **if** $r_{q,k}^2 < r_{q,k_q}^2$ **then**
      $k_q \leftarrow k$ {Update memberships}
    **end if**
  **end for**
  **for** $k = 1, \ldots, K$ **do**
    $\mathcal{K} \leftarrow \emptyset$ {Reset Clusters}
  **end for**
  **for all** $q \in \mathcal{Q}$ **do**
    $\mathcal{K}_{k_q} \leftarrow \mathcal{K}_{k_q} \cup \{q\}$
  **end for**
  **for** $k = 1, \ldots, K$ **do**
    Compute approximation $\hat{v}_k$ using samples $\mathcal{K}_k$ {Update cluster models}
    $Y_k \leftarrow \frac{1}{\#\mathcal{K}_k} \sum_{q\in\mathcal{K}_k} y^q$ {Update centroids}
  **end for**
**end while**

## 2.4 Cluster merging

Algorithm 13 sorts the observations into $K$ clusters, each hopefully belonging to region where $u$ can be accurately approximated. It may happen that more than one cluster belong to the region where a single cluster can give an accurate approximation. In such cases, merging of two clusters should be possible without a

significant increase in the approximation error.

For this purpose, we use the following iterative algorithm where at each step, we detect a couple of neighbouring clusters that can be merged. Let $K^{(l)}$ denote the number of clusters after $l-$th iteration of the procedure. For a cluster index $k < K^{(l)}$, we find the neighbouring cluster $k'$ (determined by the distance between cluster centroids) and consider the following indicator:

$$\Delta^l(k,k') = \hat{R}^2(\mathcal{K}_k \cup \mathcal{K}_{k'}) - \hat{R}^2(\mathcal{K}_k) - \hat{R}^2(\mathcal{K}_{k'}). \tag{6.12}$$

the indicator $\Delta^l(k,k')$ measures the change in the sum of squared residuals if the distinct clusters $k$ and $k'$ are merged. Here $\hat{R}^2(\mathcal{K}_k \cup \mathcal{K}_{k'})$ is the residual obtained using model $\hat{v}^{k,k'}$ constructed by merging points in clusters $\mathcal{K}_k$ and $\mathcal{K}_{k'}$. The two distinct clusters are merged if the indicator $\Delta^l(k,k') < 0$ i.e. the squared residual of the merged cluster is less than the sum of squared residuals of individual clusters $k$ and $k'$. Hence, we have $\mathcal{K}_k \leftarrow \mathcal{K}_k \cup \mathcal{K}_{k'}$ and the model $\hat{v}_k$ is updated accordingly (i.e. $\hat{v}^k \leftarrow \hat{v}^{k,k'}$). Clearly, the merging procedure stops after at most $l_f \leq K-1$ iterations and in the end we have $K^{(l_f)}$ as the final number of clusters.

## 2.5 Resampling

It remains to construct the global approximation $\hat{u}$ of $u$. Specifically, given a point in $\Xi$, we need to decide to which cluster domain it belongs. This is a classification problem and in this work we simply rely on Nearest-Neighbour strategy. Let $\mathcal{N}_n(y) \subset \mathcal{Q}$ be the set of $n$ observations the closest to $y^q$ (in the sense of Euclidean distance). One can then poll memberships $k_q$ for $q \in \mathcal{N}_n(y)$ to decide which cluster $y^q$ belongs to. We can then select the cluster having the most frequent membership among $\mathcal{N}_n(y^q)$ and eventually breaks ties by taking the membership of the closest point. In the examples in section 3, we consider $n = 5$. Let $k(\mathcal{N}_n(y))$ denote the cluster selected using the Nearest-Neighbour classification procedure in the following.
We can then consider the global approximation

$$u(y) \approx \hat{u}(y) = \sum_{k \in \{1,\ldots,K^{l_f}\}} 1_k(y)\hat{v}_k(y),\ 1_k(y) = \begin{cases} 1, \text{ if } k = k(\mathcal{N}_n(u)). \\ 0, \text{ otherwise.} \end{cases} \tag{6.13}$$

In the following section, we will illustrate this strategy on two examples.

# 3 Illustrations

## 3.1 2D vibration problem

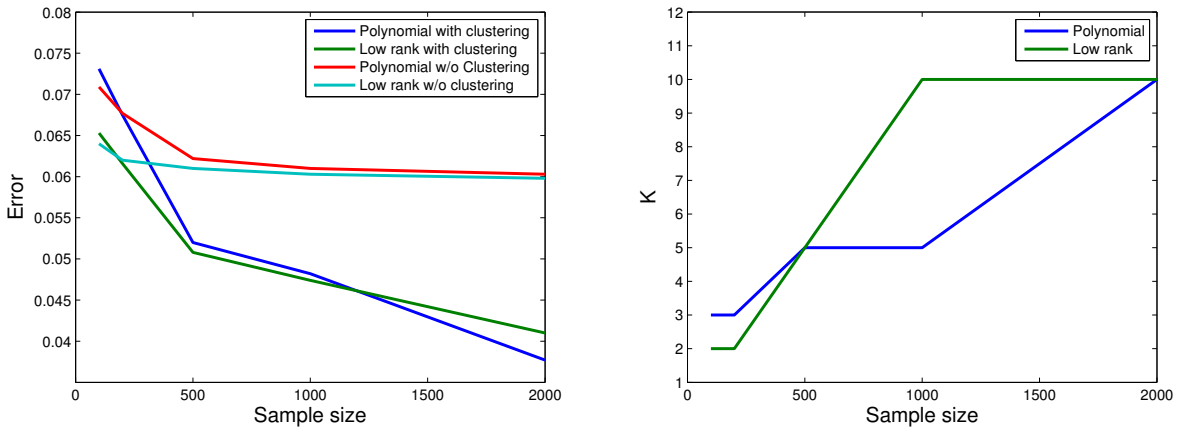
We consider a forced vibration problem of a slightly damped linear elastic beam defined on a spatial domain $\Omega = [0,1]$ where the source of uncertainty comes from

the modulus of elasticity. The problem writes: find $u(x,\xi)$ such that

$$\begin{aligned} \frac{\partial^2}{\partial x^2}\left(E(\xi)I\frac{\partial^2 u}{\partial x^2}\right) - \rho S\omega^2 u = 0 \quad \text{for } x \in \Omega, \\ u = 0,\ \frac{\partial u}{\partial x} = 0 \quad \text{at } x = 0, \\ \frac{\partial^2 u}{\partial x^2} = 0\ ,\ \frac{\partial}{\partial x}\left(E(\xi)I\frac{\partial^2 u}{\partial x^2}\right) = 1 \quad \text{at } x = 1, \end{aligned}$$

where $\rho = 7800$ is the density mass, $S$ and $I$ are respectively the area and second moment of the area of the beam's circular cross-section of radius 0.1, $E = E_0(1-i\eta)$ is the modulus of elasticity with loss factor $\eta = 0.001$ and $E_0$ modelled as a uniform random variable: $E_0 \sim U(10^{11}, 3\,10^{11})$ which is expressed as a function of a uniform random variable $\xi_1 \sim U(-1,1)$. The circular frequency $\omega$ is here considered as a parameter of the problem that varies on $[2\pi.10, 2\pi.10^4]$ and is expressed as a function of parameter $\xi_2 \in [-1,1]$. We define the set of parameters $\xi = (\xi_1,\xi_2) \in [-1,1]^2$. The variable of interest of the problem is defined as follows:

$$I(u)(\xi) = \log \|u(1,\xi)\|.$$




(a) (b)

**Figure 6.1:** 2D vibration problem: The plot in (a) compares the error obtained with and without clustering and classification approach with sample size. Plot (b) shows the number of initial clusters $K$ corresponding to minimum error with different sample sizes

**Results** We consider the approximation spaces $\mathcal{S}_n^1 = \mathcal{S}_n^2 = \mathbb{P}_{n-1}$ for $n \in \{2,\ldots,5\}$. We use sparse polynomial approximation for constructing model approximations $\hat{v}_k$. For each sample set, we consider several values of initial clusters $K \in \{2,4,5,6,10\}$

and $\gamma \in \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ and in each cluster and the optimal model (w.r.t to $K$ and $\gamma$) is selected using cross validation. In fig. 6.1(a), we compare the performance of clustering and classification approach with direct sparse polynomial approximation. We find that the error obtained with approximations using clustering and classification approach reduces with increase in sample size where as direct approximations ceases to improve. Also, for small sample size ($Q = 100, 200$), low rank approximations are more accurate as compared to polynomial approximations. We also plot initial number of cluster $K$ that corresponds to minimum error for each sample size in fig. 6.1(b). We observe that, for smaller sample sets, small values of $K$ are chosen whereas one can afford high values of $K$ for large sample sets (or for moderate number of samples when using low rank approximations). Indeed, for high sample size, enough points are available in each cluster to construct a more accurate model approximation.

## 3.2 Manhattan function

Let us consider a manufactured function (named Manhattan function) given by:

$$u(\xi) = \begin{cases} \text{checkerboard}(\xi_1, \xi_2) & 0 \leq \xi_1 \leq 1, \ -1 \leq \xi_2 \leq 1 \\ \sin(7\xi_1)\sin(4\xi_2) & -1 \leq \xi_1 < 0, \ 0 \leq \xi_2 \leq 1 \\ \frac{1}{2} + \frac{2}{7}(2\xi_1 + 1)^2 + (2\xi_2 + 1)^2 & -1 \leq \xi_1 < 0, \ -1 \leq \xi_2 < 0 \end{cases} \tag{6.14}$$

Figure 6.14 illustrates this function. In each subdomain, this function is of low rank and hence is ideal to test the proposed approach with low rank approximation. As in section 5.2, we consider piecewise polynomials of degree 2 defined on a uniform partition of $\Xi_k$ composed by 8 intervals i.e. $\mathcal{S}_n^1 = \mathcal{S}_n^2 = \mathbb{P}_{2,8}$. Note that this basis is suitable to exactly approximate the checkerboard function in (6.14). We consider approximation models $\hat{v}_k \in \mathcal{R}_{m_k}$ with optimal value of rank $m_k$ selected using cross validation. We consider $K \in \{4, 5, 6\}$ with values of $\gamma \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. Figure 6.3 illustrates the evolution of clusters at different steps of this approach using 2000 samples.

**Results:** Figure 6.3 shows the distribution of points in several clusters at different steps of the strategy. In fig 6.4(a), we compare the performance of clustering and classification approach with direct sparse low rank approximation. We find that, as in example 3.1 the error obtained with low rank approximation using clustering and classification approach reduces with increase in sample size where as the accuracy of direct sparse low rank approximation does not significantly improve with increase in sample size. In fig 6.4(b), we plot the initial cluster size $K$ and final cluster size $K^{(l_f)}$ (selected using cross validation) that corresponds to minimum error for each sample set. Again, as in previous example, we find that a more accurate model approximation is obtained if initial cluster size $K$ is more than the number of sub-domains corresponding to three conditions in (6.14). Also,

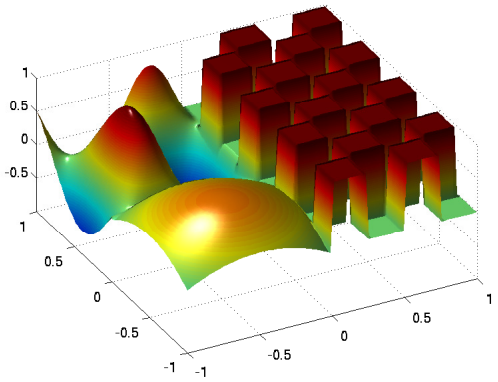



**Figure 6.2:** Manhattan function

the optimal number of final cluster $K^{(l_f)} = 3$ is obtained for large samples set ($Q > 1000$).

### 3.3 5D vibration problem

We here consider the model problem presented in section 3.1 with $E_0(x, \xi)$ and $\eta(x, \xi)$ defined as follows:

$$E_0(x, \xi_1) \sim U(1.6\,10^{11}, 2.4\,10^{11}) \text{ and } \eta(x, \xi_2) \sim U(0.001, 0.05) \text{ for } x \in [0, 0.5[,$$
$$E_0(x, \xi_4) \sim U(1.6\,10^{11}, 2.4\,10^{11}) \text{ and } \eta(x, \xi_5) \sim U(0.001, 0.05) \text{ for } x \in ]0.5, 1].$$

The circular frequency $\omega$ is a parameter that varies on $[2\pi.10, 2\pi.10^4]$ expressed as a function of parameter $\xi_3 \in [-1, 1]$. We define the set of parameters $\xi = (\xi_k)_{k=1}^5 \in [-1, 1]^5$.
The variable of interest of the problem is: $I(u)(\xi) = \log \|u(1, \xi)\|$. In order to demonstrate the irregularity of the function $u$, in fig. 6.5 we plot (logarithm of) $u$ with respect to different variables.

**Results:** We construct sparse low rank canonical models $\hat{v_k} \in \mathcal{R}_m$ in approximation space $\otimes_{i=1}^5 \mathbb{P}_5$ in each cluster. In fig. 6.6, we compare the performance of clustering and classification approach with direct sparse low rank approximation. We draw the same conclusion as in previous illustrations that the accuracy of approximation obtained with clustering and classification is better as compared to direct sparse low rank construction.

## 4 Conclusion

The clustering and classification approach offers an improvement in accuracy for discontinuous and irregular functions provided enough sample evaluations are available.

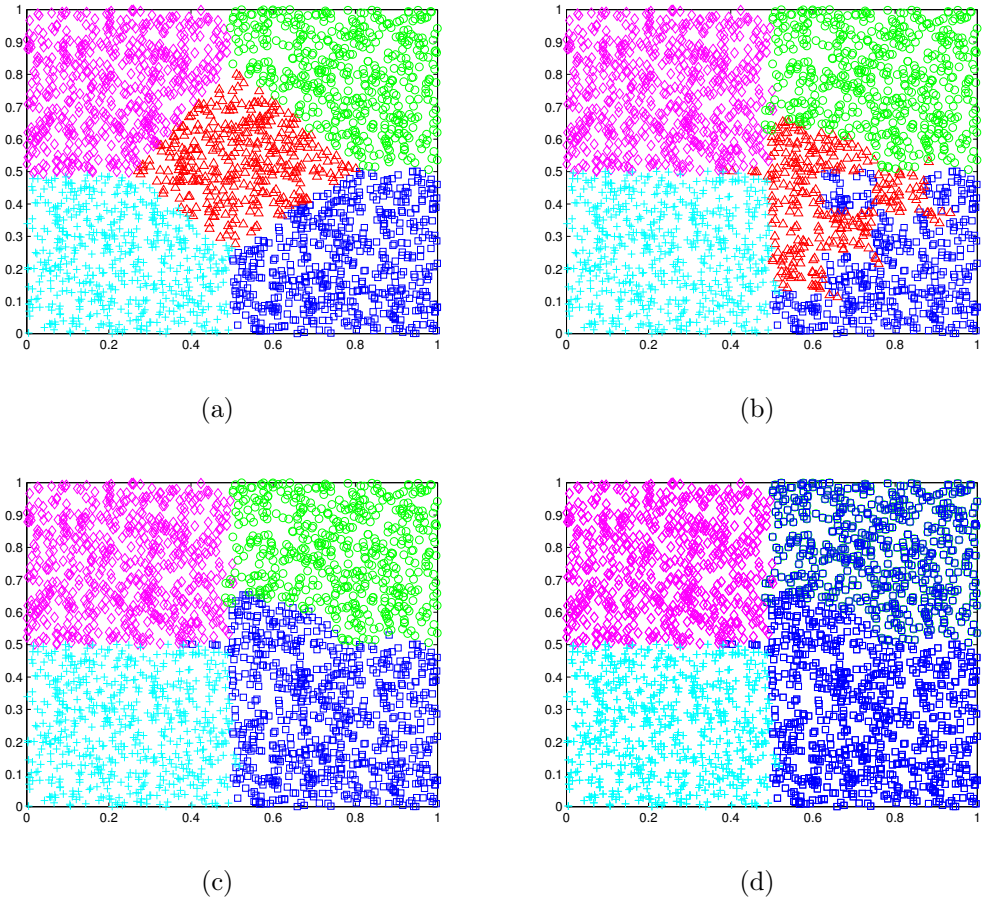
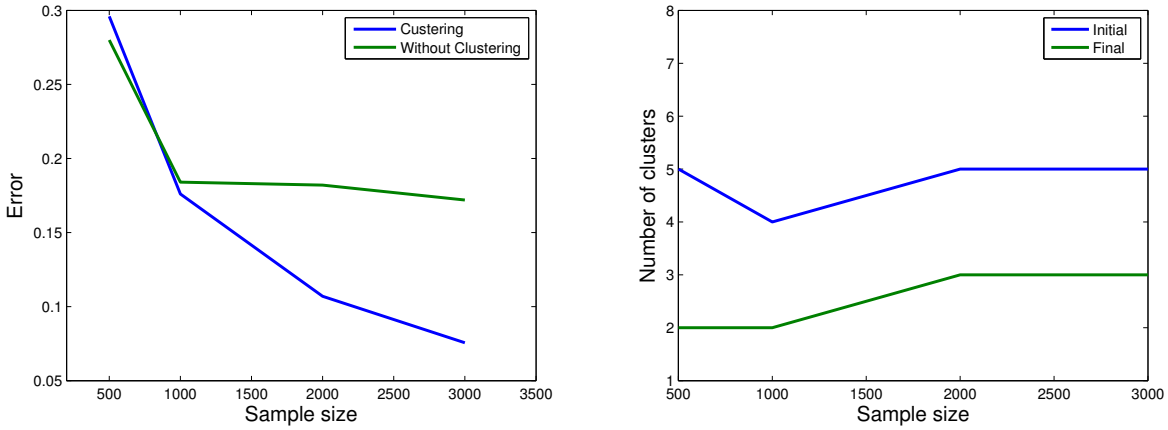


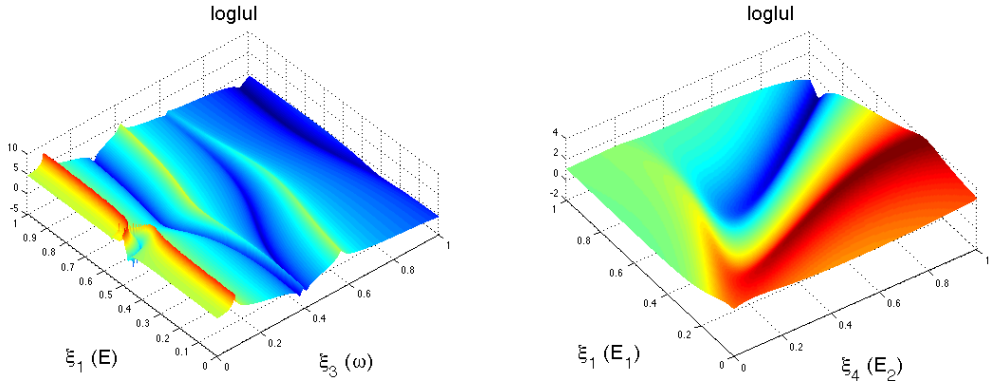
**Figure 6.3:** Manhattan function: The plot in (a) shows the initial clustering of 2000 samples into 5 clusters after application of algorithm 12. Plot (b) shows the clustering after application of algorithm 13. Plot (c) and (d) shows the merging of clusters to obtain the final clustering

The illustrations in section 3 are promising and more analysis is required to further improve this strategy. One possible improvement could be to use cross validation based error indicator to identify clusters with inaccurate models and improve their accuracy by adaptive sampling or choice of appropriate basis. A selection between different types of models (polynomial, low rank) approximation models in different clusters can give better accuracy for few samples.




(a) (b)

**Figure 6.4:** Manhattan problem: The plot in (a) compares the error obtained with and without clustering and classification approach with sample size. Plot (b) shows the number of initial cluster ($K$) and final cluster ($K^{(l_f)}$) corresponding to minimum error with different sample size




(a) (b)

**Figure 6.5:** 5D vibration problem

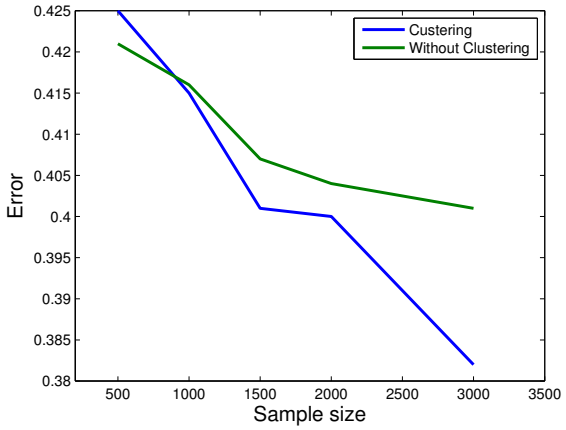



**Figure 6.6:** 5-D vibration problem: Comparison of the error obtained with and without clustering and classification approach with sample size.

# Conclusions and Perspectives

Uncertainty quantification has now emerged as an important field of research for improving the predictability and robustness of numerical models. Approaches based on functional representation gives a framework based on classical results in functional analysis. In many cases, it may not be possible to access or modify existing deterministic computational codes. In such cases, sampling based approaches provide a viable option for the resolution of uncertainty quantification problems. However, as in other functional representation based approaches, these approaches are limited to low dimensional problems due to the curse of dimensionality.

Recently, a number of methods have been introduced that exploit certain structures of high dimensional functions. One such method is based on the interpretation of high dimensional functions as tensors in a tensor product space. Such interpretation has the advantage of drawing from the recent results in low rank approximation of tensors. At the same time, methods exploiting sparsity based on compressed sensing theory has recently been introduced with encouraging results.

In this thesis, we have combined both these approaches. More specifically, we have exploited sparsity within low rank structure for high dimensional functions. It is based on the observation that a large class of functions encountered in practical applications are found to have sufficiently accurate low rank representations and hence can be parametrized with few parameters in appropriate tensor product subsets. This leads to introduce sparse low rank tensor formats. These subsets are characterised by the associated notion of rank. Algorithms based on alternating least-squares are used to perform approximation in these subsets. We introduced algorithms to perform approximation in two specific low rank tensor formats (canonical and tensor train) and illustrated their application on several examples. Finally, we introduced strategies to combine these approaches with clustering and classification techniques for approximating discontinuous and irregular functions.

This approach opens doors for further theoretical and practical developments which are outlined below:

1. The present work consisted of considering *isotropic* approximation of tensors i.e. we used the same type (polynomial, multi-element, wavelet...) and number

of function basis in each dimension. A possible improvement could be to devise strategies for adaptive selection of basis functions in each stochastic dimension.

2. The present work is based on random sampling of input random variables. Theoretical estimates of the number of samples necessary to obtain *stable* low rank approximation for a given class of functions remains an area of further research. Sampling strategies that better exploit sparsity and low rank structure can also be introduced. Also, adaptive sampling that favours better estimation of the most influential parameter or estimation of the probability of a particular event can be a subject of further improvement.

3. We can evaluate the quality of approximation of a sparse tensor subset with respect representation of function and devise strategies to select more suitable tensor subsets (selection of trees in tree based tensor format) that could be based on a general hierarchical vision of grouping certain variables which are not suitable for separation.

# Bibliography


[1] I. Babuška, F. Nobile, and R. Tempone. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM J. Num. Anal.*, 45(3):1005–1034, 2007.

[2] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

[3] Markus Bachmayr, Wolfgang Dahmen, Ronald DeVore, and Lars Grasedyck. Approximation of high-dimensional rank one tensors. *Constructive Approximation*, 39(2):385–395, 2014.

[4] G. Beylkin, B. Garcke, and M.J. Mohlenkamp. Multivariate regression and machine learning with sums of separable functions. *Journal of Computational Physics*, 230:2345–2367, 2011.

[5] G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM J. Sci. Comput.*, 26(6):2133–2159, 2005.

[6] Gregory Beylkin, Jochen Garcke, and Martin J Mohlenkamp. Multivariate regression and machine learning with sums of separable functions. *SIAM Journal on Scientific Computing*, 31(3):1840–1857, 2009.

[7] Gregory Beylkin and Martin J Mohlenkamp. Algorithms for Numerical Analysis in High Dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, January 2005.

[8] G. Blatman and B. Sudret. Sparse polynomial chaos expansions and adaptive stochastic finite elements using a regression approach. *Comptes Rendus Mécanique*, 336(6):518–523, 2007.

[9] G. Blatman and B. Sudret. Adaptive sparse polynomial chaos expansion based least angle regression. *Journal of Computational Physics*, 230:2345–2367, 2011.

[10] H-J Bungartz and Stefan Dirnstorfer. Multivariate quadrature on adaptive sparse grids. *Computing*, 71(1):89–114, 2003.

[11] H-J. Bungartz and M. Griebel. Sparse grids. *Acta. Numer.*, 13:147–269, 2004.

[12] R. E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta. Numer.*, 7:1–49, 1998.

[13] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *Information Theory, IEEE Transactions on*, 52(12):5406–5425, 2006.

[14] J. Tao T. Candes, EJ. Romberg. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on information theory*, 52(12):5406–5425, 2006.

[15] Gavin C Cawley and Nicola LC Talbot. Fast exact leave-one-out cross-validation of sparse least-squares support vector machines. *Neural networks*, 17(10):1467–1475, 2004.

[16] Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

[17] Francisco Chinesta, Pierre Ladeveze, and Elías Cueto. A short review on model order reduction based on proper generalized decomposition. *Archives of Computational Methods in Engineering*, 18(4):395–404, 2011.

[18] Abdellah Chkifa, Albert Cohen, Ronald DeVore, and Christoph Schwab. Sparse adaptive taylor approximation algorithms for parametric and stochastic elliptic pdes. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(01):253–280, 2013.

[19] Abdellah Chkifa, Albert Cohen, Giovanni Migliorati, Fabio Nobile, and Raul Tempone. Discrete least squares polynomial approximation with random evaluations; application to parametric and stochastic elliptic pdes. *Also available as EPFL-MATHICSE report*, pages 35–2013, 2013.

[20] Abdellah Chkifa, Albert Cohen, and Christoph Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric pdes. *Foundations of Computational Mathematics*, pages 1–33, 2013.

[21] Abdellah Chkifa, Albert Cohen, and Christoph Schwab. Breaking the curse of dimensionality in sparse polynomial approximation of parametric pdes. *Journal de Mathématiques Pures et Appliquées*, 2014.

[22] P. Constantine, E. Dow, and Q. Wang. Active subspace methods in theory and practice: applications to kriging surfaces. *arXiv preprint arXiv:1304.2070*, 2013.

[23] L. De Lathauwer, B. De Moor, and J. Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21(4):1253–1278, 2000.

[24] V. de Silva and L.-H. Lim. Tensor rank and ill-posedness of the best low-rank approximation problem. *SIAM Journal of Matrix Analysis & Appl.*, 30(3):1084–1127, 2008.

[25] Vin de Silva and Lek-Heng Lim. Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, January 2008.

[26] M. Deb, I. Babuška, and J. T. Oden. Solution of stochastic partial differential equations using galerkin finite element techniques. *Computer Methods in Applied Mechanics and Engineering*, 190:6359–6372, 2001.

[27] D.L. Donoho. Compressed Sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[28] A. Doostan and G. Iaccarino. A least-squares approximation of partial differential equations with high-dimensional random inputs. *Journal of Computational Physics*, 228(12):4332–4345, 2009.

[29] A. Doostan and H. Owhadi. A non-adapted sparse approximation of pdes with stochastic inputs. *Journal of Computational Physics*, 230(8):3015–3034, 2011.

[30] Bradley Efron, Trevor Hastie, Iain Johnstone, Robert Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[31] Lars Eldén and Berkant Savas. A newton-grassmann method for computing the best multilinear rank-(r_1, r_2, r_3) approximation of a tensor. *SIAM Journal on Matrix Analysis and applications*, 31(2):248–271, 2009.

[32] Mike Espig and Wolfgang Hackbusch. A regularized newton method for the efficient approximation of tensors represented in the canonical tensor format. *Numerische Mathematik*, 122(3):489–525, 2012.

[33] C.G. Webster F. Nobile, R. Tempone. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46(5):2309–2345, 2007.

[34] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numer. Algorithms*, 18:209–232, 1998.

[35] R. Ghanem and P. Spanos. *Stochastic finite elements: a spectral approach.* Springer, Berlin, 1991.

[36] Loïc Giraldi. *Contributions aux méthodes de calcul basées sur l'approximation de tenseurs et applications en mécanique numérique*. PhD thesis, Ecole centrale de nantes-ECN, 2012.

[37] L. Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.*, 31:2029–2054, 2010.

[38] Lars Grasedyck, Daniel Kressner, and Christine Tobler. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013.

[39] W. Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*, volume 42 of *Series in Computational Mathematics*. Springer, 2012.

[40] Wolfgang Hackbusch. *Tensor Spaces and Numerical Tensor Calculus*. Springer, 2012.

[41] Wolfgang Hackbusch and Stefan Kühn. A new scheme for the tensor representation. *Journal of Fourier Analysis and Applications*, 15(5):706–722, 2009.

[42] Aicke Hinrichs, Erich Novak, Mario Ullrich, and Henryk Woźniakowski. The curse of dimensionality for numerical integration of smooth functions ii. *Journal of Complexity*, 30(2):117–143, 2014.

[43] Toshimitsu Homma and Andrea Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1):1–17, 1996.

[44] Mariya Ishteva, P-A Absil, and Paul Van Dooren. Jacobi algorithm for the best low multilinear rank approximation of symmetric tensors. *SIAM Journal on Matrix Analysis and Applications*, 34(2):651–672, 2013.

[45] Mariya Ishteva, Lieven De Lathauwer, P-A Absil, and Sabine Van Huffel. Differential-geometric newton method for the best rank-(r 1, r 2, r 3) approximation of tensors. *Numerical Algorithms*, 51(2):179–194, 2009.

[46] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, September 2009.

[47] Daniel Kressner and Christine Tobler. Preconditioned low-rank methods for high-dimensional elliptic pde eigenvalue problems. *Computational Methods in Applied Mathematics Comput. Methods Appl. Math.*, 11(3):363–381, 2011.

[48] O. P. Le Maître, O. M. Knio, H. N. Najm, and R. Ghanem. Uncertainty propagation using Wiener-Haar expansions. *Journal of Computational Physics*, 197:28–57, 2004.

[49] O.P. Le Maître, O. M. Knio, H. N. Najm, and R. G. Ghanem. Uncertainty propagation using Wiener-Haar expansions. *Journal of Computational Physics*, 197(1):28–57, 2004.

[50] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.

[51] L. Mathelin and O. Le Maître. Dual-based a posteriori error estimate for stochastic finite element methods. *Communications in Applied Mathematics and Computational Science*, 2(1):83–116, 2007.

[52] Hermann G Matthies, Alexander Litvinenko, Oliver Pajonk, Bojana V Rosić, and Elmar Zander. Parametric and uncertainty computations with tensor product representations. In *Uncertainty Quantification in Scientific Computing*, pages 139–150. Springer, 2012.

[53] H. Niederreiter. *Random Number Generation and quasi-Monte Carlo Methods*. SIAM, Philadelphia, PA, 1992.

[54] Anthony Nouy. Proper generalized decompositions and separated representations for the numerical solution of high dimensional stochastic problems. *Archives of Computational Methods in Engineering*, 17(4):403–434, 2010.

[55] E. Novak and K. Ritter. Simple cubature formulas with high polynomial exactness. *Constr. Approx.*, 15:499–522, 1999.

[56] Erich Novak and Daniel Rudolf. Tractability of the approximation of high-dimensional rank one tensors. *arXiv preprint arXiv:1402.5011*, 2014.

[57] Erich Novak and Henryk Woźniakowski. *Tractability of Multivariate Problems: Standard information for functionals*, volume 2. European Mathematical Society, 2010.

[58] IV Oseledets. Constructive representation of functions in low-rank tensor formats. *Constructive Approximation*, 37(1):1–18, 2013.

[59] IV Oseledets and EE Tyrtyshnikov. Tensor tree decomposition does not need a tree. *preprint*, 8, 2009.

[60] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[61] Ivan V Oseledets and Eugene E Tyrtyshnikov. Breaking the curse of dimensionality, or how to use svd in many dimensions. *SIAM Journal on Scientific Computing*, 31(5):3744–3759, 2009.

[62] Núria Parés, Pedro Díez, and Antonio Huerta. Bounds of functional outputs for parabolic problems. part i: Exact bounds of the discontinuous galerkin time discretization. *Computer Methods in Applied Mechanics and Engineering*, 197(19):1641–1660, 2008.

[63] K. Petras. Smolyak cubature of given polynomial degree with few nodes for increasing dimension. *Numer. Math.*, 93:729–753, 2003.

[64] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, Cambridge, 2nd edition, 1997.

[65] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011.

[66] G. I. Schüeller and P. D. Spanos (eds). *Monte Carlo Simulation*. Balkema, Rotterdam, 2001.

[67] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Sov. Math. Dokl*, 3:240–243, 1963.

[68] Ilya M Sobol? Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and computers in simulation*, 55(1-3):271–280, 2001.

[69] Ilya M Sobol. Theorems and examples on high dimensional model representation. *Reliability Engineering & System Safety*, 79(2):187–193, 2003.

[70] Il'ya Meerovich Sobol'. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, 2(1):112–118, 1990.

[71] I.M. Sobol. On quasi-monte carlo integrations. *Math. Comput. Simulat.*, 47:103–112, 1998.

[72] Christian Soize and Roger Ghanem. Physical systems with random uncertainties: chaos representations with arbitrary probability measure. *SIAM Journal on Scientific Computing*, 26(2):395–410, 2004.

[73] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.

[74] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[75] R. Tipireddy and R. Ghanem. Basis adaptation in homogeneous chaos spaces. *J.Comp.Phy*, 259:304–317, 2014.

[76] Christine Tobler. *Low-rank tensor methods for linear systems and eigenvalue problems*. PhD thesis, Diss., Eidgenössische Technische Hochschule ETH Zürich, Nr. 20320, 2012, 2012.

[77] Nick Vannieuwenhoven, Raf Vandebril, and Karl Meerbergen. A new truncation strategy for the higher-order singular value decomposition. *SIAM Journal on Scientific Computing*, 34(2):A1027–A1052, 2012.

[78] X. Wan and G.E. Karniadakis. An adaptive multi-element generalized polynomial chaos method for stochastic diffential equations. *J. Comp. Phys.*, 209:617–642, 2005.

[79] X. Wan and G.E. Karniadakis. Multi-element generalized polynomial chaos for arbitrary propability measures. *SIAM J. Sci. Comp.*, 28(3):901–928, 2006.

[80] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863, 1992.

[81] Brian A Worley. Deterministic uncertainty analysis. Technical report, Oak Ridge National Lab., TN (USA), 1987.

[82] D. Xiu and G. E. Karniadakis. The Wiener-Askey polynomial chaos for stochastic differential equations. *SIAM J. Sci. Comput.*, 24(2):619–644, 2002.